BabyLM 2025

The First BabyLM Workshop: Accelerating Language Modeling Research with Cognitively Plausible Data

Proceedings of the Workshop

November 8, 2025

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-355-5

Introduction

We are excited to welcome attendees to the First BabyLM Workshop! This follows two years of the BabyLM Challenge (now in its third iteration). The workshop will be co-located with the 2025 Conference on Empirical Methods in Natural Language Processing on November 8, 2025 in Suzhou, China.

This year, the program includes an oral session for the winning shared task papers, two oral sessions for the award-winning workshop papers, and a poster session for all accepted submissions. There is also an introductory presentation from the organizers summarizing the challenge, this year's winning submissions, and trends across submissions. In addition, we have two invited talks—one from a language modeling expert and another from a cognitive modeling expert.

We received 32 workshop submissions (many of which also included system submissions to the BabyLM Challenge) and 12 direct challenge submissions. We are grateful to the challenge participants, whether in the challenge or workshop tracks, for advancing the science of language modeling. The participants' efforts are essential to advancing the state of cognitively plausible and sample-efficient language modeling.

We also extend our thanks to the organizers of EMNLP for their significant efforts in sustaining a conference of its scale, and in providing an environment for the BabyLM community.

Finally, we thank our program committee members—largely sampled from the participants of the challenge—for committing their time to help us curate an excellent program.

—The BabyLM Organizing Committee: Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Y. Hu, Jing Liu, Jaap Jumelet, Tal Linzen, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Gotlieb Wilcox, Adina Williams

Organizing Committee

Program Chairs

Lucas Charpentier, University of Oslo
Leshem Choshen, Massachusetts Institute of Technology
Ryan Cotterell, ETH Zürich
Mustafa Omer Gul, Cornell University
Michael Y. Hu, New York University
Jing Liu, École Normale Supérieure - PSL
Jaap Jumelet, University of Groningen
Tal Linzen, New York University
Aaron Mueller, Boston University
Candace Ross, Meta AI
Raj Sanjay Shah, Georgia Tech
Alex Warstadt, UC San Diego
Ethan Wilcox, Georgetown University
Adina Williams, Meta AI

Program Committee

BabyLM Challenge Reviewers

Lisa Beinborn, Alessandro Bondielli

Luca Capone

Lukas Edman

Achille Fusco

Nalin Kumar

Mateusz Lango, Sharid Loáiciga

Francesca Padovani, Alexandros Potamianos

Asad B. Sayeed, Jun Suzuki

Joonas Tapaninaho

BabyLM Workshop Reviewers

Xuanda Chen

Justin DeBenedetto

Steven Y. Feng

Kata Gabor, Zebulon Goriely, Leon Guertler

Patrick Haller, Michael Y. Hu

Mayank Jobanputra

Marianne De Heer Kloots, Andrey Kutuzov

Mateusz Lango, Yukyung Lee, Jing Liu, Sharid Loáiciga

Bhavitvya Malik

Ercong Nie

Miyu Oba, Yohei Oseki

Omkar Pangarkar

Anthony Rios

David Samuel, Raj Sanjay Shah, Ekta Sood, Shikhar Srivastava, Julius Steuer, Michal Štefánik

Alex Warstadt

Han Yang

Li Zhou, Richard Zhu

Keynote Talk

Benchmarking Baby and Large Language Models in Chinese

Hai Hu
City University of Hong Kong
2025-11-08 11:00:00

Abstract: In this talk, I will first briefly overview the efforts in creating linguisticsally oriented benchmarks in Chinese. I will discuss the design and construction of benchmarks targeting the orthorgraphy, phonology, syntax, logic and semantics, pragmatics and world knowledge of modern and classical Chinese, by our lab and other teams in the field. The evaluations of baby and large language models show that current LLMs are very powerful, especially with the addition of "reasoning" abilities. However, certain linguistic blindspots remain and further refinement of evaluation tasks and methodologies is needed. Next, I will discuss ongoing studies in understanding the learning mechanisms of monolingual and bilingual language models involving Chinese. Finally I will point out what the LM community might learn from the language acquisition community.

Keynote Talk

Small Language Models through Insights from Human Language Acquisition

Yohei Oseki
University of Tokyo
2025-11-08 15:30:00

Abstract: Large language models (LLMs) have achieved remarkable success, thanks to the rapid development of AI and machine/deep learning, and outperformed humans at various downstream tasks. However, those LLMs, despite their super-human performance, have been pointed out as not efficient in terms of training data, model parameters, and computational resources. In this talk, I propose small language models (SLMs) that efficiently learn natural language like humans, building on insights from human language acquisition. Specifically, SLMs are trained on developmentally plausible corpora like BabyLM Challenge via curriculum learning, batch learning, direct/indirect evidence, variation set, and critical period. The results suggest that inductive biases are essential to efficiently train SLMs, with scientific implications for human language acquisition, as well as engineering applications to edge AI and low-resource languages.

Table of Contents

BabyLM Workshop

BabyLM Challenge

Program

Saturday, November 8, 2025

09:00 - 09:15 *Opening Remarks*

09:15 - 10:15 *BabyLM Challenge Orals*

CLASS-IT: Conversational and Lecture-Aligned Small-Scale Instruction Tuning for BabyLMs
Luca Capone, Alessandro Bondielli and Alessandro Lenci

Masked Diffusion Language Models with Frequency-Informed Training
Despoina Kosmopoulou, Efthymios Georgiou, Vaggelis Dorovatas, Georgios Paraskevopoulos and Alexandros Potamianos

MoEP: Modular Expert Paths for Sample-Efficient Language Modeling
Joonas Tapaninaho

Mask and You Shall Receive: Optimizing Masked Language Modeling For Pretraining BabyLMs
Lukas Edman and Alexander Fraser

Once Upon a Time: Interactive Learning for Storytelling with Small Language Models
Jonas Mayer Martins, Ali Hamza Bashir, Muhammad Rehan Khalid and Lisa Beinborn

10:15 - 11:00 *Break*

11:00 - 12:00 *Invited Talk 1: Hai Hu*

12:00 - 13:30 *Lunch*

13:30 - 15:00 *Poster Session*

15:00 - 15:30 *Break*

15:30 - 16:30 *Invited Talk 2: Yohei Oseki*

16:30 - 17:05 *BabyLM Workshop Orals*

Teacher Demonstrations in a BabyLM's Zone of Proximal Development for Contingent Multi-Turn Interaction
Suchir Salhan, Hongyi gu, Donya Rooein, Diana Galvan-Sosa, Gabrielle Gaudeau, Andrew Caines, Zheng Yuan and Paula Buttery

Are BabyLMs Deaf to Gricean Maxims? A Pragmatic Evaluation of Sample-efficient Language Models
Raha Askari, Sina Zarrieß, Özge Alacam and Judith Sieker

Looking to Learn: Token-wise Dynamic Gating for Low-Resource Vision-Language Modelling
Bianca-Mihaela Ganescu, Suchir Salhan, Andrew Caines and Paula Buttery

17:05 - 17:15 *Awards and Closing Remarks*

Rethinking the Role of Text Complexity in Language Model Pretraining

Dan John Velasco* and **Matthew Theodore Roque***
Samsung R&D Institute Philippines
{dj.velasco,roque.mt}@samsung.com
***Equal Contribution**

Abstract

Improving pretraining data quality and size is known to boost downstream performance, but the role of text complexity—how hard a text is to read—remains less explored. We reduce surface-level complexity (shorter sentences, simpler words, simpler structure) while keeping core content approximately constant and ask: (i) How does complexity affect language modeling across model sizes? (ii) Can useful representations be learned from simpler text alone? (iii) How does pretraining text complexity influence downstream language understanding? We simplify human-written texts using a large language model, pretrain causal models (28M–500M) from scratch on original vs. simplified data, and evaluate them in fine-tuning and zero-shot setups. We find that perplexity is sensitive to the interaction between model capacity and text complexity—smaller models degrade far less on simpler texts—while text complexity has little impact on fine-tuning evaluations, with zero-shot evaluations indicating that simpler texts benefit performance on linguistic knowledge tasks, whereas more complex texts favor tasks requiring world knowledge and entity tracking. Our findings suggest that different types of data diversity affect transfer and zero-shot performance differently, providing insight into tailoring data curation to specific goals.

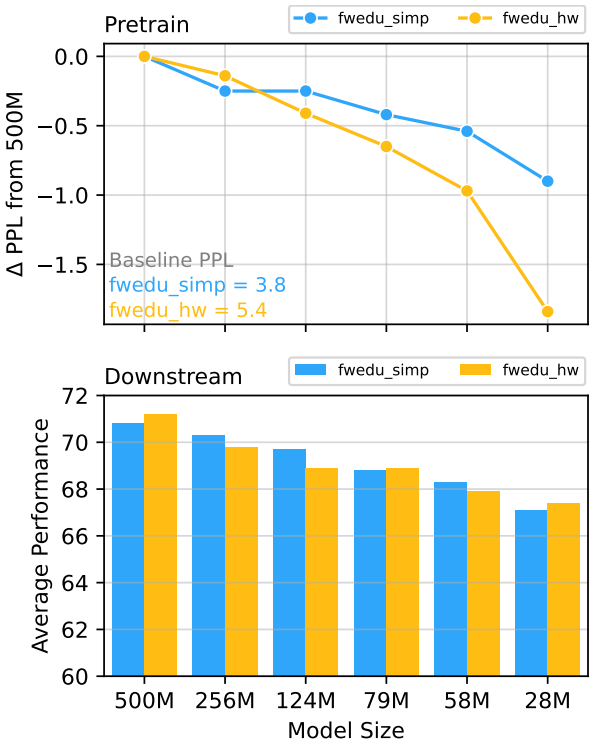


Figure 1: **(Top)** Perplexity (PPL) degrades faster for models trained on `fwedu_hw` (human-written) than on `fwedu_simp` (simplified) as model size decreases, suggesting that smaller models handle lower-complexity text more effectively. **(Bottom)** Average performance across 7 language tasks remains similar across data setups suggesting text complexity has limited impact on general language understanding.

1 Introduction

Let's compare two versions of text:

(A) As the sunset cast its warm orange glow over Manila Bay, people relaxed on the sideline benches, enjoying the peaceful view of the sunset.

(B) The sunset gave Manila Bay a warm, orange light. People sat on the benches and enjoyed the view of the sunset.

The two versions convey the same core meaning, but one uses more nuanced, complex language, whereas the other is simpler and less nuanced. This can be likened to lossy compression, where version (B) requires fewer bits to represent the information in (A) but loses some nuance. It compresses by using common words and simpler sentence structures while retaining the core information.

Proceedings of the First BabyLM Workshop, pages 1–28
November 8, 2025

What if our corpus is more like (B)? Can we still learn useful representations by training solely on simplified text with a simpler vocabulary and sentence structure? To answer this, we manipulate surface-level complexity—shorter sentences, simpler words, simpler structures—while keeping the semantics close to constant, and measure downstream performance.

It is well-known that language models acquire world knowledge during pretraining (Petroni et al., 2019; Roberts et al., 2020; Zhang et al., 2021; Wei et al., 2022), and transfer learning is more effective when the pretraining corpus aligns with the target task domain (Ruder and Plank, 2017; Gururangan et al., 2020). For example, pretraining on medical-related texts leads to better performance on medical domain tasks than using finance-related texts. This highlights that a model's knowledge base strongly affects downstream results. To isolate the effect of text complexity, it's essential to control for core content. In this paper, we ask three core questions:

(1) How does text complexity affect language modeling performance across models of varying capacity?

(2) Can we learn useful representations by training solely on simpler text, with simpler vocabulary and sentence structure?

(3) How does the text complexity of pretraining data affect downstream performance on language understanding tasks?

We collected human-written texts and used a Large Language Model (LLM) to produce simplified versions while preserving core content. Causal language models (28M-500M) were then pretrained from scratch in two setups: on the original texts and on their simplified counterparts. We evaluated language understanding through fine-tuning, and linguistic knowledge and commonsense reasoning in zero-shot settings.

Our empirical evidence shows that reducing surface-level complexity features does not significantly impact performance on general language understanding tasks (Figure 1). These results suggest that text complexity is not the primary driver of performance; instead, knowledge coverage may matter more. However, zero-shot evaluations (Table 4 and 5) suggest that text simplicity can boost performance in linguistic knowledge tasks, while greater complexity tends to aid world knowledge and entity tracking.

2 Related Work

Text complexity or readability. It refers to how difficult a text is to understand (DuBay, 2004), influenced by linguistic factors such as word choice (e.g., "utilize" vs. "use"), sentence structure (complex vs. simple), and content type (academic vs. children's books) (Dale and Chall, 1948; Graesser et al., 2004). Other factors such as the reader's knowledge affect readability (Ozuru et al., 2009). In this work, we focus solely on linguistic aspects.

Common readability metrics—such as Flesch Reading Ease (FRE) (Flesch, 1948), Dale-Chall (Dale and Chall, 1948), and SMOG (Mc Laughlin, 1969)—use surface features like sentence length and word complexity. These measures overlook deeper dimensions such as coherence and style, motivating machine learning and deep learning approaches (Hancke et al., 2012; Meng et al., 2020; Imperial, 2021; Chatzipanagiotidis et al., 2021). Recent work applies LLMs to readability estimation (Trott and Rivière, 2024; Lee and Lee, 2023; Rooein et al., 2024), achieving strong alignment with human judgments even without fine-tuning. However, LLM-based scoring is computationally costly at corpus scale, so we use FRE to estimate readability.

Text simplification (TS). It aims to make text easier to understand while preserving its content (Agrawal and Carpuat, 2023; Alva-Manchego et al., 2019; Truică et al., 2023). While simplified texts tend to be shorter, this is not always the case (Shardlow, 2014). This is different from Text Summarization, where the goal is to shorten the text even if it changes the organization and content. Early approaches used word substitution with lexicons (Saggion and Hirst, 2017; Shardlow, 2014; Kriz et al., 2018), while others framed TS as statistical machine translation (SMT) (Wubben et al., 2012; Scarton et al., 2018; Specia, 2010; Xu et al., 2016). Subsequent work applied deep learning encoder-decoder models (Zhang and Lapata, 2017; Alva-Manchego et al., 2019; Agrawal and Carpuat, 2023), and recent studies explore LLMs (Trott and Rivière, 2024; Imperial and Tayyar Madabushi, 2023; Farajidizaji et al., 2024; Padovani et al., 2024). While some research targets specific grade levels, we follow Trott and Rivière (2024) in simplifying complex texts without grade constraints, leveraging LLMs for this task.

Pretraining language models on simple texts. Recent work has explored pretraining small language models (SLMs) on simple texts. Huebner et al. (2021) showed that models trained on child-directed speech match larger models on probing tasks. Eldan and Li (2023) found that SLMs trained on synthetic short stories using only words familiar to 3-4-year-olds can generate coherent, fluent text. Other studies (Deshpande et al., 2023; Muckatira et al., 2024) reported that SLMs pretrained on simplified language perform comparably to larger models when problems are reformulated in simpler terms. The BabyLM Challenge (Warstadt et al., 2023; Hu et al., 2024) pretrains language models on <100M words from child-directed and simplified texts, using provided or custom datasets within the budget.

Pretraining dataset design. Large-scale pretraining is a key driver of modern language model performance (Brown et al., 2020; Kaplan et al., 2020; Hoffmann et al., 2022). Dataset design choices—domain composition, quality and toxicity filtering, and collection date—affect performance in ways that fine-tuning cannot fully correct (Longpre et al., 2024).

The work most related to ours, Agrawal and Singh (2023), finds that models trained on more complex text (e.g., Wikipedia) outperform those trained on simpler text (e.g., children's books), with complexity measured by Flesch Reading Ease. However, because they compare entirely different corpora, complexity is confounded with other factors such as topic breadth, register, discourse structure, and domain diversity. We instead manipulate complexity within the same source texts, preserving core content while varying only surface-level features. This controlled setup isolates the effect of textual complexity, complementing the broader correlation observed by Agrawal and Singh (2023).

Although prior work reports positive results for simple-text pretraining, no study has directly examined the impact of surface-level complexity at larger scales (e.g., 2B tokens). Our experiments address this gap, providing empirical evidence on whether useful models can be trained solely on simplified text.

3 Creating the Pretraining Datasets

3.1 Human-Written Corpus

We took a subset of FineWeb-Edu (Penedo et al., 2024), a collection of high-quality English web pages specifically optimized for educational content. This dataset is known for its permissive license[1] and its quality since it has gone through rigorous processing such as filtering, deduplication, and curation. The subset has 2 billion tokens[2], denoted as **`fwedu_hw`** (short for FineWeb-Edu human-written). The choice of dataset size is motivated by Chinchilla Compute-Optimal guideline of 1:20 parameter-tokens ratio (Hoffmann et al., 2022) and practical reasons (e.g. training under fixed compute budget). While this size is not exactly compliant to the Chinchilla guideline for the 124M, 256M and 500M models, it is at least Chinchilla Optimal for the smaller models (e.g. 28M, 58M, 79M).

3.2 Simplified Corpus

We prompt Llama 3.1 8B (Grattafiori et al., 2024) to transform `fwedu_hw` into simplified texts. For efficient inference, we use the INT8 quantized version[3] of the model and vLLM (Kwon et al., 2023) as our LLM serving system. We discuss more about the prompt engineering and include the final prompt in Appendix B.

We split the documents from `fwedu_hw` into paragraphs[4]. Transformation is done at the **paragraph level** because the model tends to summarize rather than simplify if the input is a multi-paragraph document. However, not all paragraphs are transformed. This can happen under three conditions: (1) when a paragraph is too short relative to its full document (e.g. title, headers); (2) when a paragraph is too long (e.g. tables, lists); or (3) when the transformation is significantly shorter or longer than the original text (e.g. incorrect simplification). In all of these cases, we **removed these paragraphs from both datasets**. This allows for more control on text complexity of the datasets while controlling for text content by keeping both datasets perfectly parallel. On both datasets, the paragraphs were not reconstructed back to document-level and instead, pretraining is done at the paragraph-level. We include a more detailed breakdown of these conditions in Appendix C.

[1] ODC-By 1.0

[2] Token counts derived from Llama 2 tokenizer (Touvron et al., 2023)

[3] Model accessed at: `https://huggingface.co/neuralmagic/Meta-Llama-3.1-8B-Instruct-quantized.w8a8`

[4] We use the term "paragraph" to refer to the smallest unit of block of text in our data pipeline. It is **not** always the case that the smallest unit is an actual paragraph. It can be a single sentence, table, heading, author lists, or other text artifacts.

The final simplified corpus, denoted as **fwedu_simp** (short for FineWeb-Edu simplified), has around 1.71B tokens. To get a rough idea of what the simplified texts look like, see the following example:

> **Original**: Your comment really helped me feel better the most. I was sitting in my office, feeling so bad that I didn't say how inappropriate and out of line his comments were, and this helped.
>
> **Simplified**: Your comment really helped me feel better. I was feeling bad because I didn't speak up when someone made inappropriate comments.

4 Experimental Setup

In our study, we investigate the effect of text complexity on pretraining and downstream performance of language models across varying model capacity. We compare models trained on `fwedu_hw` (human-written) with those trained on `fwedu_simp` (simplified).

4.1 Model Architecture

Our model architecture is based on the design choices of MobileLLM (Liu et al., 2024): deep-and-thin architectures, SwiGLU activation (Shazeer, 2020), grouped-query attention (Ainslie et al., 2023), and embeddings sharing (Press and Wolf, 2017). All models share the same set of architectural details and hyperparameters as MobileLLM except where explicitly varied. We removed embeddings sharing for the sole purpose of making the results more generalizable to most contemporary causal language models which do not use embeddings sharing. Architectural details are summarized in Table 1.

4.2 Pretraining Configurations

All models are trained for one epoch on either `fwedu_hw` or `fwedu_simp`. Both use the LLaMA-2 BPE tokenizer (Touvron et al., 2023) with a 32k vocabulary. Training examples are individual paragraphs, with no concatenation or sequence packing.

Inputs are right-padded to 512 tokens with the EOS token (identical to BOS) and use a causal attention mask. Each corpus is trained on independently and remains perfectly parallel after filtering, ensuring differences in model behavior stem solely from surface-level complexity.

#Params (Non-Emb)	#Layer	#Head	#KV	Emb Dim	#Params
500M	40	18	6	1044	~531M
256M	30	9	3	846	~283M
124M	30	9	3	576	~143M
79M	30	9	3	450	~94M
58M	30	9	3	378	~70M
28M	30	9	3	252	~36M

Table 1: Model architecture configurations. Emb Dim is the embedding size, Non-Emb refers to non-embedding parameters, and #KV denotes key-value heads. The number of layers and attention heads is fixed for models up to 256M and increased for the 500M model. This design maintains a consistent deep-and-thin architecture while scaling parameter count.

Optimization uses AdamW (Loshchilov and Hutter, 2019) with default hyperparameters, a peak learning rate of $3e{-}4$ (28M models) or $5e{-}4$ (all other models), linearly decayed, 5% warm-up, and no dropout. Models with 28M-124M parameters use an effective batch size of 256 (8 examples/GPU $\times$ 8 GPUs $\times$ 4 gradient accumulation steps). Models with 256M-500M parameters use 4 examples/GPU with 8 accumulation steps to match the batch size. Training is performed in FP16 mixed precision on 8$\times$ NVIDIA P100 GPUs; gradient checkpointing is enabled only for the 500M model.

Validation is run on a held-out corpus slice after 300M tokens and at every subsequent doubling. Results are from the final checkpoint. Implementation uses PyTorch and Hugging Face Transformers. All runs fix the random seed to 42 for data shuffling and initialization.

4.3 Fine-tuning Tasks

To assess the downstream impact of text complexity during pretraining, we fine-tune our models on a suite of seven language understanding tasks drawn from GLUE and SuperGLUE: BoolQ, MNLI, MRPC, MultiRC, QQP, RTE, and WSC. This set follows the evaluation configuration of the BabyLM Challenge, and we use the same preprocessed datasets provided in the BabyLM evaluation pipeline for both training and validation (Charpentier et al., 2025).

All models are trained with an added classification head. For tasks involving multiple input sequences (e.g., premise-hypothesis or sentence pairs), we concatenate the two sequences with a separator token before feeding them into the model. We perform two fine-tuning regimes for

each model-task pair: (1) full-model fine-tuning, where all model parameters and the classification head are updated; and (2) linear probing, where only the classification head is updated.

Fine-tuning uses 8×P100 GPUs without gradient accumulation. Batch size is determined by GPU memory constraints: for BoolQ and MultiRC we use 2 examples per GPU (effective batch size of 16), and for all other tasks we use 8 examples per GPU (effective batch size of 64). For each task, we perform a grid search over learning rates $1e{-}4, 5e{-}5, 2e{-}5, 1e{-}5, 5e{-}6$ and training epochs $1, 2, 3, 4, 5$.

We use the same task metrics as the BabyLM evaluations, namely: 3-class accuracy for MNLI; binary accuracy for BoolQ, MultiRC, and WSC; and F1-Score for MRPC and QQP. All experiments are run with three random seeds; we report the mean and standard deviation of the best-performing configuration for each seed. Fine-tuning is performed for all model sizes on all tasks under both training regimes.

4.4 Zero-shot Tasks

To further evaluate the quality of representations, we conduct zero-shot evaluations on eight multiple-choice benchmarks, grouped into two categories:

Linguistic knowledge, entity tracking, and world knowledge: BLiMP (Warstadt et al., 2020) and the BLiMP Supplement (as provided in the BabyLM evaluation pipeline), probe syntactic and morphological phenomena. Entity Tracking (Kim and Schuster, 2023) and EWoK (Elements of World Knowledge Ivanova et al., 2024), measure a model's ability to follow discourse entities and recall factual knowledge. The models were evaluated on these tasks using the BabyLM evaluation pipeline (Charpentier et al., 2025).

Commonsense reasoning: ARC-Easy and ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), and PIQA (Bisk et al., 2020), require reasoning over everyday scenarios and physical commonsense. The models were evaluated on these tasks using the `lm-evaluation-harness` (Gao et al., 2024).

All tasks are multiple-choice. For each evaluation instance, we format the input according to the benchmark's specifications and score each candidate option by the sum of log-probabilities of its tokens given the prompt. The option with the highest score is selected as the model's prediction.

We report accuracy for all zero-shot tasks. Evaluation is deterministic, as predictions depend solely on model likelihoods and not on sampling.

5 Results

We performed three independent runs with different random seeds. For each run, we selected the best result over our fixed hyperparameter grid, and report the average of those three best scores. Random seeds were fixed for full reproducibility.

5.1 Dataset Complexity Verification

Is the simplified corpus truly simpler? To answer, we compute per-dataset and cross-dataset metrics (Table 2) and analyzed their distributions (Figure 2). The simplified corpus has fewer tokens, a smaller vocabulary (Types), lower lexical diversity (Type-Token Ratio), and reduced unpredictability (Unigram Entropy)—all indicating lower text complexity. Cross-dataset metrics show that 26.62% of the data are more concise, 92% exhibits low to medium lexical overlap (ROUGE-2), and 79% retains at least 80% semantic similarity (Cosine Similarity). These results suggest that the simplified dataset **differs in form while preserving core content**.

Feature	Simplified	Human-written
PER-DATASET STATS		
Total tokens	1.71B	2.00B
Total words	1.44B	1.57B
Types (unique words)	2.76M	5.23M
Type-token ratio (%)	0.19%	0.33%
Unigram entropy (bits)	9.87	10.58
CROSS-DATASET STATS		
Compression (<80%)	26.62%	—
Exact match	2.51%	—
High lexical overlap	6.47%	—
Medium lexical overlap	31.13%	—
Low lexical overlap	61.75%	—
Exact mismatch	1.56%	—
Semantic Sim (>80%)	79.00%	—

Table 2: Per-dataset and Cross-dataset statistics. Reduced per-dataset stats in Simplified indicate lower complexity compared to Human-written. Lexical overlap is measured using ROUGE-2 (R2), with the following thresholds: exact match ($R2 = 1$), high ($0.8 < R2 < 1$), medium ($0.4 < R2 \leq 0.8$), low ($0 < R2 \leq 0.4$), and exact mismatch ($R2 = 0$). Semantic Sim is computed as cosine similarity of paragraph embeddings. Cross-dataset stats suggest Simplified texts differ in form but preserve core content.

Figure 2 compares the distributions of paired metrics for `fwedu_simp` and `fwedu_hw`, labeled as "Paired". The first-row metrics are adapted

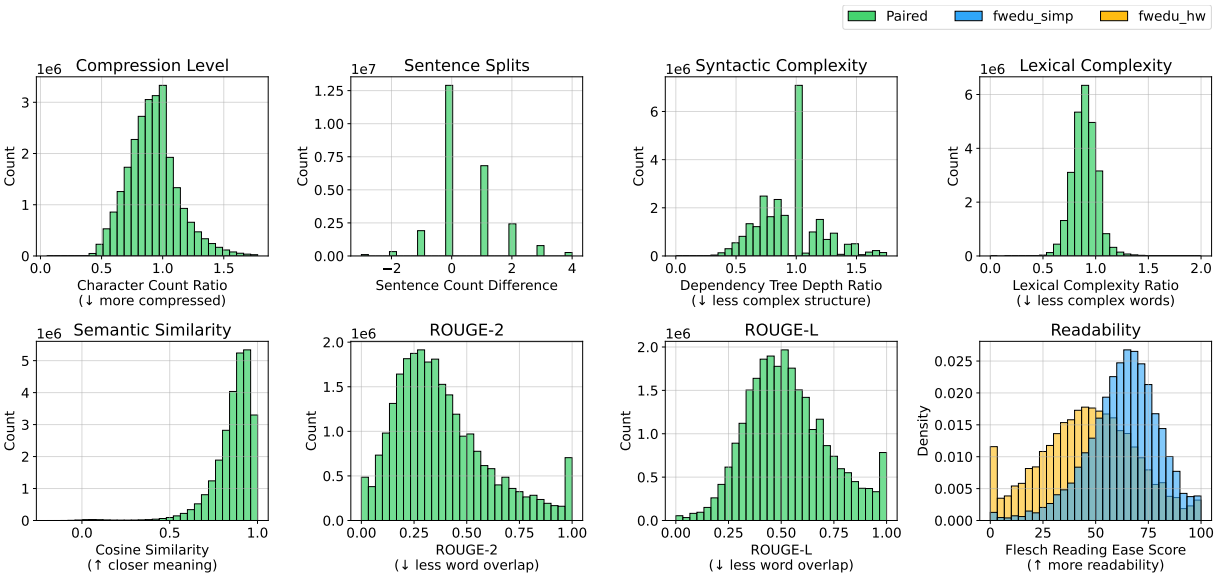


Figure 2: Corpus Features distribution. First row shows metrics of `fwedu_simp` to `fwedu_hw`. Second row are pairwise metrics except for Flesch Reading Ease (FRE) which only requires one input. The first row suggests `fwedu_simp` is shorter, has more sentences, uses simpler structures, and more common words. The second row shows that `fwedu_simp` is semantically similar to `fwedu_hw`, with low word-order overlap (low ROUGE-2), moderate preservation of idea flow and structure (moderate ROUGE-L), and clearly higher FRE, indicating systematic differences in readability. For visualization, we removed outliers, which account for only 2.9% of the data (see Appendix F.2 for definition and examples of outliers).

from ASSET's text complexity evaluation (Alva-Manchego et al., 2020):

- **Compression Level**: ratio of character counts; values < 1.0 indicate more concise texts.
- **Sentence Splits**: difference in sentence counts; values > 0 indicate splitting of complex sentences into simpler ones.
- **Syntactic Complexity**: ratio of maximum dependency-tree depth; values < 1.0 indicate shallower (simpler) sentence structures.
- **Lexical Complexity**: mean squared log-rank of non-stopword tokens, based on the top 50k words in FastText embeddings[5] (Mikolov et al., 2018); values < 1.0 indicate use of more frequent words.

The second row of Figure 2 shows semantic similarity, word overlap, and readability. We used `all-MiniLM-L6-v2` to encode paragraph embeddings, optimized for tasks like sentence similarity and clustering[6]. High scores near 1.0 indicate most simplified paragraphs retain the original meaning.

ROUGE was computed with the Evaluate tool (Von Werra et al., 2022), scoring from 0 to 1. ROUGE-2 measures exact bigram overlap, reflecting local phrasing; most examples score 0-0.4, indicating low lexical overlap. This confirms high cosine similarity reflects shared meaning, not surface form. ROUGE-L, measuring longest in-order subsequences, shows more varied scores, suggesting moderate structural similarity.

Readability was measured using Flesch Reading Ease (FRE), which factors text length, word count, and syllables. Higher FRE means simpler text: easy (60+), fairly difficult (50-60), hard (<50) (Scott, 2024). `fwedu_hw` skews lower, `fwedu_simp` higher, indicating systematic readability differences.

Together, these results show our simplified dataset is **simpler in form while preserving core content**. For examples, see Appendix D and E.

5.2 Main Comparison: Human-Written vs. Simplified

5.2.1 Language-Modeling Performance

How does text complexity affect language modeling performance across models of varying capacity? To answer this, we measured each model's perplexity degradation—defined as the absolute dif-

[5] 2 million word vectors trained on Common Crawl (600B tokens), `https://fasttext.cc/docs/en/english-vectors.html`

[6] `all-MiniLM-L6-v2` ranks 1st and 17th on the MTEB leaderboard (Muennighoff et al., 2023) for <100M and <1B parameter models.

Model	boolq	mnli	mrpc	multirc	qqp	rte	wsc	Avg.
Majority Baseline	64.0	33.1	68.1	57.5	62.7	53.9	61.5	57.3
28M								
from_scratch	66.9 ± 0.8	36.0 ± 0.5	70.4 ± 1.0	59.0 ± 0.3	71.3 ± 0.3	58.3 ± 2.1	65.5 ± 2.1	61.1
fwedu_hw	69.9 ± 0.2	61.9 ± 0.7	79.0 ± 1.7	59.1 ± 0.2	78.6 ± 0.3	60.6 ± 2.6	62.5 ± 4.7	67.4
fwedu_simp	69.3 ± 0.6	61.8 ± 0.2	78.2 ± 1.5	58.8 ± 0.2	78.8 ± 0.2	60.0 ± 2.0	63.1 ± 2.7	67.1
58M								
from_scratch	67.5 ± 0.7	37.7 ± 1.2	70.8 ± 0.7	58.8 ± 0.4	71.8 ± 0.3	57.4 ± 2.9	64.9 ± 1.0	61.3
fwedu_hw	69.6 ± 0.4	63.2 ± 0.5	79.6 ± 1.5	58.5 ± 0.3	79.6 ± 0.1	60.2 ± 1.4	64.3 ± 1.8	67.9
fwedu_simp	69.6 ± 0.1	63.6 ± 0.2	77.9 ± 1.0	58.2 ± 0.4	80.2 ± 0.0	66.4 ± 2.4	61.9 ± 5.5	68.3
79M								
from_scratch	67.4 ± 0.8	38.7 ± 0.3	70.0 ± 0.3	58.3 ± 0.3	72.1 ± 0.4	56.2 ± 3.7	65.5 ± 1.0	61.2
fwedu_hw	69.9 ± 0.3	64.0 ± 0.3	79.5 ± 1.1	58.7 ± 0.2	80.4 ± 0.4	66.2 ± 2.1	63.7 ± 4.1	68.9
fwedu_simp	69.3 ± 1.0	65.2 ± 0.3	80.3 ± 1.4	58.2 ± 0.4	80.7 ± 0.1	67.4 ± 2.5	60.7 ± 0.0	68.8
124M								
from_scratch	68.0 ± 0.8	39.5 ± 1.4	70.8 ± 0.7	58.3 ± 0.5	71.9 ± 0.1	57.4 ± 4.2	65.5 ± 2.1	61.6
fwedu_hw	70.7 ± 0.2	65.9 ± 0.2	80.3 ± 1.4	59.2 ± 0.0	80.7 ± 0.2	65.5 ± 3.2	60.1 ± 1.0	68.9
fwedu_simp	70.3 ± 0.4	66.9 ± 0.5	82.1 ± 0.7	58.8 ± 0.3	81.2 ± 0.1	67.6 ± 1.7	61.3 ± 2.7	69.7
256M								
from_scratch	68.0 ± 0.7	40.5 ± 0.9	71.0 ± 1.0	58.5 ± 0.7	72.1 ± 0.2	57.6 ± 1.0	65.2 ± 1.3	61.8
fwedu_hw	70.8 ± 0.4	67.9 ± 0.4	80.6 ± 2.0	59.1 ± 0.2	81.4 ± 0.2	66.0 ± 0.7	62.5 ± 1.8	69.8
fwedu_simp	70.8 ± 0.7	67.0 ± 0.5	81.7 ± 1.0	58.6 ± 0.4	81.5 ± 0.1	70.8 ± 0.7	61.3 ± 2.1	70.3
500M								
from_scratch	68.6 ± 0.3	39.6 ± 0.4	72.4 ± 0.3	58.5 ± 0.0	72.5 ± 0.0	60.6 ± 2.0	65.5 ± 1.0	62.5
fwedu_hw	70.5 ± 0.5	67.6 ± 0.1	83.4 ± 1.0	58.8 ± 0.2	82.1 ± 0.0	71.5 ± 0.7	64.3 ± 3.1	71.2
fwedu_simp	70.1 ± 0.5	67.4 ± 0.5	82.7 ± 0.7	58.7 ± 0.3	81.6 ± 0.1	71.8 ± 1.1	63.1 ± 2.1	70.8

Table 3: Full fine-tuning performance on 7 NLU tasks. Average accuracy across tasks is reported over 3 runs. The Avg. column reports mean accuracy over available tasks. Task metrics are as follows: 3-class accuracy (MNLI), Binary Accuracy (BoolQ, MultiRC, WSC), and F1-Score (MRPC, QQP). Overall results shows minimal performance difference across pretraining setups regardless of model size which suggests text complexity have minimal impact on general language understanding tasks.

ference in perplexity between the 500M model and smaller models. Figure 1 shows that models trained on `fwedu_hw` degrade faster than those trained on `fwedu_simp` as capacity decreases, with a sharp drop for the 28M model on `fwedu_hw`. This interaction between model capacity and data complexity suggests that future model design and selection should account for the complexity of the training data.

5.2.2 Fine-tuning Evaluation

The results of the full-model fine-tuning are summarized in Table 3. To contextualize the impact of pretraining, we include scores from the Majority baseline and models fine-tuned from random weights (`from_scratch`). Notably, MNLI shows the greatest gain from pretraining. More broadly, similar `from_scratch` performance across tasks and model sizes suggests an upper bound imposed by the training data on this specific architecture—model scaling alone does not improve performance. On MultiRC, all models perform only slightly above the majority baseline, suggesting failure to learn the task. We suspect this stems from paragraph-level pretraining, which may lack signals for skills like coreference resolution which is important to succeed in MultiRC. The same likely applies to WSC.

Overall, `fwedu_hw` and `fwedu_simp` yield similar performance across model sizes in full-model fine-tuning. This pattern holds under linear probing as well (Table 6 in Appendix A), reinforcing the observation. Full-model fine-tuning also confirms the well-established trend that larger models perform better, regardless of data complexity. These results suggest that text complexity is not the primary driver of performance; instead, knowledge coverage may matter more.

5.2.3 Zero-shot Evaluation

Linguistic Knowledge, Entity Tracking, and World Knowledge. Table 4 summarizes the zero-shot performance on linguistic knowledge and entity tracking benchmarks. BLiMP performance improves with model size, with both setups performing similarly overall, though `fwedu_simp` has a clear edge on BLiMP-supplement. Entity tracking performance varies widely with model size; `fwedu_hw` often leads, while `fwedu_simp` surpasses random chance (20%) only from 79M on-

Model	blimp	blimp-supp	ewok	entity
28M				
fwedu_hw	67.83	55.90	52.79	16.15
fwedu_simp	66.90	57.47	52.67	18.78
58M				
fwedu_hw	69.69	59.03	52.69	26.35
fwedu_simp	70.73	62.15	53.81	18.36
79M				
fwedu_hw	70.67	60.47	54.09	28.89
fwedu_simp	70.44	61.55	53.09	20.73
124M				
fwedu_hw	71.64	62.61	54.07	25.09
fwedu_simp	71.30	63.27	54.01	21.79
256M				
fwedu_hw	72.37	62.61	56.18	29.71
fwedu_simp	72.53	63.65	55.09	22.58
500M				
fwedu_hw	72.23	61.85	56.72	20.81
fwedu_simp	72.60	63.79	54.85	22.05

Table 4: Zero-shot evaluations on grammatical knowledge (blimp), world knowledge (ewok), and Entity Tracking (entity) show consistent improvement with model size. Both setups perform similarly on BLiMP, `fwedu_simp` scores higher on BLiMP-supplement, whereas `fwedu_hw` leads on Entity and slightly on EWoK.

Model	arc_e	arc_chl	hellaswag	piqa
28M				
fwedu_hw	33.33	20.22	26.99	56.20
fwedu_simp	31.94	21.59	26.40	55.93
58M				
fwedu_hw	39.10	19.54	27.38	57.89
fwedu_simp	33.59	21.42	27.50	58.22
79M				
fwedu_hw	38.80	21.25	27.69	58.71
fwedu_simp	38.05	20.65	27.43	59.52
124M				
fwedu_hw	38.09	19.37	28.08	58.16
fwedu_simp	38.85	21.50	28.36	60.77
256M				
fwedu_hw	42.09	22.70	28.85	60.99
fwedu_simp	38.80	20.82	28.94	61.10
500M				
fwedu_hw	40.99	21.25	28.30	58.16
fwedu_simp	33.96	18.43	27.52	57.99

Table 5: Zero-shot accuracy on commonsense reasoning benchmarks shows that ARC-Challenge (arc_chl) remains near random chance (20%) across all setups. All other tasks improve consistently with model size. `fwedu_hw` performs best on ARC-Easy (arc_e), while `fwedu_simp` slightly outperforms on PIQA. Both setups perform similarly on HellaSwag. All 500M models show a performance drop relative to 256M across tasks.

ward. EWoK performance improves consistently with model size, with `fwedu_hw` often slightly outperforming `fwedu_simp`.

Commonsense Reasoning. Table 5 summarizes zero-shot performance on commonsense reasoning benchmarks. Performance generally improves with increased model size, especially on ARC-Easy and PIQA. ARC-Challenge remains difficult across all setups, with accuracies near random chance (20%). This may be simply due to the pretraining data not containing the knowledge that ARC-Challenge is designed to test. On ARC-Easy, `fwedu_hw` consistently outperforms `fwedu_simp`, reaching a peak accuracy of 42.09% at 256M—3 points higher than `fwedu_simp`. On Hellaswag, both setups perform comparably across model sizes. While for PIQA, `fwedu_simp` slightly outperforms `fwedu_hw` consistently. Interestingly, 500M models perform worse than 256M models across tasks. We suspect this is due to the limited data size—2B tokens for `fwedu_hw` and 1.71B for `fwedu_simp`—bottlenecking the larger models.

6 Discussion

In this section, we reflect on the broader implications of our findings for data curation and synthetic data generation. We frame these as conjectures rather than definitive claims.

Our experiments controlled for lexical, syntactic, semantic, and stylistic diversity, though these do not exhaust the full space of variation. Ideally, optimizing along multiple dimensions may yield broader benefits, but practical constraints often force trade-offs. Our results provide empirical evidence on which outcomes, such as transfer or zero-shot performance, are most sensitive to particular forms of diversity. This can help guide decisions when prioritizing which dimensions to optimize.

Data curation. In data curation or pruning, practitioners sometimes emphasize surface-level variety (lexical or syntactic) as a proxy for diversity. Our experiments suggest this can be misleading. We find that reducing lexical and syntactic variation, while preserving topical and knowledge coverage, did not harm transfer performance but did impair zero-shot generalization. This implies that curation strategies should be designed with the intended use case in mind: fine-tuned applications may tolerate reduced surface variation, whereas zero-shot settings are more sensitive to it.

Synthetic data design. Generation-based synthetic datasets often suffer from low-diversity outputs (Gandhi et al., 2024) and are vulnerable to collapse effects (Shumailov et al., 2024; Guo et al., 2024; Briesch et al., 2024), underscoring the need for diversity-aware generation. Our findings indicate that not all forms of diversity contribute equally: different axes influence different outcomes. A practical design strategy may be to prioritize broad topic and knowledge coverage first, then deliberately introduce surface-level variety (e.g., controlled paraphrasing) to support zero-shot performance if needed.

7 Conclusion

In this work, we investigate how text complexity affects language model pretraining. Specifically, we ask whether simplified language—while preserving core content—can lead to representations that perform as well as those learned from more complex, human-written text. We pretrained causal language models of varying sizes (28M-500M parameters) on both simplified and human-written corpora. Our results show that simplifying surface-level features does not significantly hurt downstream performance on a range of language understanding tasks. However, models trained on more complex text show an advantage in zero-shot settings on benchmarks requiring reasoning and knowledge of the world—such as Entity Tracking, EWoK, and ARC-Easy—while performing similarly on BLiMP, HellaSwag, and PIQA. These findings highlight that different types of data diversity affect transfer and zero-shot performance differently, providing insight into tailoring data curation to specific goals.

Limitations

Our study has several limitations. First, the LLM-based simplification process is imperfect and may introduce subtle inconsistencies in core content due to hallucinations. Second, the 2B-token corpora are relatively small by today's pretraining standards, potentially limiting model performance. Third, our fine-tuning evaluation focuses on a narrow set of classification and multiple-choice benchmarks, which may not capture the full range of model capabilities, particularly in open-ended or generative tasks. Fourth, our zero-shot evaluation may not fully reflect the targeted capabilities, as it is constrained by limited training data and model capacity. Fifth, we focus solely on causal language models, leaving open the possibility that different patterns may emerge with encoder models like BERT. Lastly, we did not conduct a per-phenomenon analysis of BLiMP, leaving open the possibility that certain linguistic constructions are more sensitive to simplification.

References

Ameeta Agrawal and Suresh Singh. 2023. Corpus complexity matters in pretraining language models. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 257–263, Toronto, Canada (Hybrid). Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2023. Controlling pre-trained language models for grade-specific text simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12807–12819, Singapore. Association for Computational Linguistics.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019. Cross-sentence transformations in text simplification. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 181–184, Florence, Italy. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Martin Briesch, Dominik Sobania, and Franz Rothlauf. 2024. Large language models suffer from their own output: An analysis of the self-consuming training loop. *Preprint*, arXiv:2311.16822.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens

Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Savvas Chatzipanagiotidis, Maria Giagkou, and Detmar Meurers. 2021. Broad linguistic complexity analysis for greek readability classification. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–58.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Vijeta Deshpande, Dan Pechi, Shree Thatte, Vladislav Lialin, and Anna Rumshisky. 2023. Honey, I shrunk the language: Language model behavior at reduced scale. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5298–5314, Toronto, Canada. Association for Computational Linguistics.

William H DuBay. 2004. The principles of readability. *Impact Information*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2024. Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9325–9339, Torino, Italia. ELRA and ICCL.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. 2024. Better synthetic data by retrieving and transforming existing datasets. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6453–6466, Bangkok, Thailand. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. The language model evaluation harness.

Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu,

Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2024. The curious decline of linguistic

diversity: Training language models on synthetic text. *Preprint*, arXiv:2311.09807.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability classification for german using lexical, syntactic, and morphological features. In *Proceedings of COLING 2012*, pages 1063–1080.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. *Preprint*, arXiv:2412.05149.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Joseph Marvin Imperial. 2021. Bert embeddings for automatic readability assessment. *arXiv preprint arXiv:2106.07935*.

Joseph Marvin Imperial and Harish Tayyar Madabushi. 2023. Flesch or fumble? evaluating readability standard alignment of instruction-tuned language models. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 205–223, Singapore. Association for Computational Linguistics.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyurek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. 2018. Simplification using paraphrases and context-based lexical substitution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 207–217.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Bruce W. Lee and Jason Lee. 2023. Prompt-based learning for text readability assessment. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1819–1824, Dubrovnik, Croatia. Association for Computational Linguistics.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. *Preprint*, arXiv:2402.14905.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Changping Meng, Muhao Chen, Jie Mao, and Jennifer Neville. 2020. Readnet: A hierarchical transformer framework for web article readability analysis. In *Advances in Information Retrieval: 42nd European*

Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42, pages 33–49. Springer.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Sherin Muckatira, Vijeta Deshpande, Vladislav Lialin, and Anna Rumshisky. 2024. Emergent abilities in reduced-scale generative language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1242–1257, Mexico City, Mexico. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Yasuhiro Ozuru, Kyle Dempsey, and Danielle S McNamara. 2009. Prior knowledge, reading skill, and text cohesion in the comprehension of science texts. *Learning and instruction*, 19(3):228–242.

Francesca Padovani, Caterina Marchesi, Eleonora Pasqua, Martina Galletti, and Daniele Nardi. 2024. Automatic text simplification: A comparative study in Italian for children with language disorders. In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 176–186, Rennes, France. LiU Electronic Press.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Preprint*, arXiv:2406.17557.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Donya Rooein, Paul Röttger, Anastassia Shaitarova, and Dirk Hovy. 2024. Beyond flesch-kincaid: Prompt-based metrics improve difficulty classification of educational texts. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 54–67, Mexico City, Mexico. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Horacio Saggion and Graeme Hirst. 2017. *Automatic text simplification*, volume 32. Springer.

Carolina Scarton, Gustavo Paetzold, and Lucia Specia. 2018. Text simplification from professionally produced corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Bryan Scott. 2024. Learn about the flesch reading ease formula.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2024. The curse of recursion: Training on generated data makes models forget. *Preprint*, arXiv:2305.17493.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *Computational Processing of the Portuguese Language: 9th International Conference, PROPOR 2010, Porto Alegre, RS, Brazil, April 27-30, 2010. Proceedings 9*, pages 30–39. Springer.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu,

Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Sean Trott and Pamela Rivière. 2024. Measuring and modifying the readability of English texts with GPT-4. In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 126–134, Miami, Florida, USA. Association for Computational Linguistics.

Ciprian-Octavian Truică, Andrei-Ionuţ Stan, and Elena-Simona Apostol. 2023. Simplex: a lexical text simplification architecture. *Neural Computing and Applications*, 35(8):6265–6280.

Leandro Von Werra, Lewis Tunstall, Abhishek Thakur, Sasha Luccioni, Tristan Thrush, Aleksandra Piktus, Felix Marty, Nazneen Rajani, Victor Mustar, and Helen Ngo. 2022. Evaluate & evaluation on the hub: Better best practices for data and model measurements. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 128–136, Abu Dhabi, UAE. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1112–1125, Online. Association for Computational Linguistics.

Model	boolq	mnli	mrpc	multirc	qqp	rte	wsc	Avg.
Baseline								
majority	64.0	33.1	68.1	57.5	62.7	53.9	61.5	57.3
28M								
fwedu_hw	65.1 ± 0.9	42.9 ± 0.3	69.9 ± 0.7	57.2 ± 0.6	69.7 ± 0.3	53.9 ± 2.6	50.6 ± 5.7	58.5
fwedu_simp	65.7 ± 1.6	43.0 ± 2.2	70.7 ± 1.0	54.8 ± 0.5	69.8 ± 0.2	56.0 ± 3.4	59.5 ± 3.7	59.9
58M								
fwedu_hw	64.4 ± 0.5	45.2 ± 0.5	69.4 ± 0.6	54.8 ± 0.6	69.2 ± 0.2	49.5 ± 2.9	62.5 ± 4.7	59.3
fwedu_simp	64.2 ± 0.5	45.6 ± 0.3	69.2 ± 0.0	54.8 ± 0.2	68.1 ± 0.6	56.2 ± 1.8	56.5 ± 9.2	59.2
79M								
fwedu_hw	66.0 ± 0.3	44.4 ± 1.2	70.7 ± 0.8	55.9 ± 0.3	69.0 ± 0.5	55.3 ± 3.6	55.4 ± 8.9	59.5
fwedu_simp	64.5 ± 0.3	46.8 ± 0.5	69.7 ± 1.0	55.8 ± 1.6	68.5 ± 0.2	53.2 ± 1.4	57.1 ± 3.1	59.4
124M								
fwedu_hw	64.5 ± 0.2	45.2 ± 0.4	70.5 ± 1.5	54.2 ± 0.7	69.9 ± 0.6	50.5 ± 4.0	57.1 ± 4.7	58.9
fwedu_simp	64.5 ± 0.1	46.3 ± 0.2	70.0 ± 0.3	54.8 ± 0.4	70.3 ± 0.8	53.0 ± 3.3	57.1 ± 4.7	59.4
256M								
fwedu_hw	65.1 ± 0.1	48.3 ± 0.5	69.7 ± 0.8	56.1 ± 0.9	68.8 ± 0.2	55.8 ± 1.7	60.7 ± 3.6	60.6
fwedu_simp	65.8 ± 1.0	48.9 ± 0.8	70.0 ± 0.6	55.3 ± 1.3	70.1 ± 0.2	57.2 ± 2.4	57.1 ± 3.6	60.6
500M								
fwedu_hw	64.9 ± 0.7	49.1 ± 0.7	70.7 ± 0.5	56.0 ± 0.9	70.1 ± 0.5	49.5 ± 2.4	63.1 ± 2.7	60.5
fwedu_simp	65.0 ± 0.2	48.9 ± 0.4	71.3 ± 1.0	56.0 ± 1.7	69.2 ± 0.6	57.6 ± 6.7	59.5 ± 6.8	61.1

Table 6: Linear Probe performance on 7 NLU tasks. Average accuracy across tasks is reported over 3 runs. The Avg. column reports mean accuracy over available tasks. Overall results shows minimal performance differences between pretraining setups across model sizes. This supports the Full fine-tuning findings (Table 3), suggesting that text complexity has limited impact on general language understanding tasks.

A Linear Probing Results

B Text Simplification Prompt

The prompt engineering is done through trial-and-error and judged by the authors according to the following qualitative criteria:

- Does it use simpler words? By "simpler words," we mean commonly used words.
- Does it convert compound or complex sentences into simple sentences?
- Does it preserve the original content and organization of thoughts?

Once we found a prompt that can reliably do all those things on a small sample, we used that prompt to transform the whole corpus.

The final prompt is shown below:

```
---

Role Description:
You are an experienced educator and linguist
    specializing in simplifying complex
    texts without losing any key information
     or changing the content. Your focus is
    to make texts more accessible and
    readable for primary and secondary
    school students, ensuring that the
    essential information is preserved while
     the language and structure are adapted
    for easier comprehension.
```

```
---

Task Instructions:
1. Read the Following Text Carefully:
   - Thoroughly understand the content,
       context, and purpose of the text to
       ensure all key information is
       retained in the simplified version.

2. Simplify the Text for Primary/Secondary
    School Students:
   - Rewrite the text to make it more
       accessible and easier to understand.
   - Use age-appropriate language and simpler
        sentence structures.
   - Maintain all key information and do not
       omit any essential details.
   - Ensure that the original meaning and
       intent of the text remain unchanged.

3. Preserve Key Information:
   - Identify all essential points, facts,
       and ideas in the original text.
   - Ensure these elements are clearly
       presented in the simplified version.

4. Avoid Adding Personal Opinions or
    Interpretations:
   - Do not introduce new information or
       personal views.
   - Focus solely on simplifying the original
        content.

---

Simplification Guidelines:

Sentence Structure:
- Use simple or compound sentences.
- Break down long or complex sentences into
    shorter ones.
- Ensure each sentence conveys a clear idea.

Vocabulary:
- Use common words familiar to primary and
    secondary school students.
- Replace advanced or technical terms with
    simpler synonyms or provide brief
    explanations.
- Avoid jargon unless it is essential, and
    explain it if used.

Clarity and Coherence:
- Organize the text logically with clear
    paragraphs.
- Use transitional words to connect ideas
    smoothly.
- Ensure pronouns clearly refer to the
    correct nouns to avoid confusion.
- Eliminate redundancies and unnecessary
    repetitions.

Tone and Style:
- Maintain a neutral and informative tone.
- Avoid overly formal language.
- Write in the third person unless the text
    requires otherwise.

---

Output Format:
Provide the simplified text in clear, well-
    organized paragraphs.
Do not include the original text in your
    output.
Do not add any additional commentary or notes
    .
Ensure the final output is free of
    grammatical errors and is easy to read.
Output $<|eot_id|>$ right after the
    simplified text.

---

Example Simplifications:

Example 1:

Original Text:
"Photosynthesis is the process by which green
     plants and some other organisms use
    sunlight to synthesize foods from carbon
     dioxide and water. Photosynthesis in
    plants generally involves the green
    pigment chlorophyll and generates oxygen
     as a byproduct."

Simplified Text:
"Photosynthesis is how green plants make food
     using sunlight, carbon dioxide, and
    water. They use a green substance called
     chlorophyll, and the process produces
    oxygen.$<|eot_id|>$"


Example 2:

Original Text:
"Global warming refers to the long-term rise
    in the average temperature of the Earth'
    s climate system, an aspect of climate
    change shown by temperature measurements
     and by multiple effects of the warming
    ."

Simplified Text:
"Global warming means the Earth's average
    temperature is increasing over a long
    time. This is part of climate change and
     is shown by temperature records and
    various effects.$<|eot_id|>$"


Example 3:

Original Text:
"The mitochondrion, often referred to as the
    powerhouse of the cell, is a double-
    membrane-bound organelle found in most
    eukaryotic organisms, responsible for
    the biochemical processes of respiration
     and energy production through the
    generation of adenosine triphosphate (
    ATP)."

Simplified Text:
"A mitochondrion is a part of most cells that
     acts like a powerhouse. It has two
    membranes and makes energy for the cell
```

```
    by producing something called ATP.$<|
    eot_id|>$"

---

Text to Simplify:
<Insert Text Here>

---

Your Output:
```

C Skipping or Rejecting Simplification

We choose to tag as `to_skip` or `to_reject` the simplification step under the following conditions: (1) the paragraph is too short relative to its full document; (2) the paragraph is too long; or (3) the transformation is significantly shorter or longer than the original text.

Condition (1) is based on two key observations. First, some textual artifacts, like titles and author names, don't require simplification. Second, very short inputs often trigger text completion instead of simplification. For example, the input "MAHATMA GANDHI" generates a passage about the person rather than a simplified version. To handle such cases, we use heuristics to determine whether a document or paragraph should be tagged as `to_skip`. First, we apply a hard rule: a document is tagged as `to_skip` if there is only one paragraph or the minimum paragraph length is greater than or equal to the standard deviation of paragraph token counts within a document. Otherwise, each paragraph in the document is evaluated based on two criteria: it is tagged as `to_skip` if it contains **10 or fewer space-separated words** or if its **token count falls below the quantile threshold of 0.15**. Meaning, paragraphs with token counts below the 15th percentile will be tagged as `to_skip`.

Condition (2) is based on the observation that paragraphs exceeding **1,500 tokens** tend to be structured texts like tables, name lists, or tables of contents, which do not need simplification. To handle such cases, we simply skip the paragraph if it exceeds 1,500 tokens. While quantile heuristics could be used, we chose the simpler heuristic.

Condition (3) is motivated by two observations. First, we observed that when asked to simplify a long input, the model tends to summarize it, significantly shortening the text and losing its original structure. Second, the model tends to append extra text, such as explanations after the answer. To detect cases where the output is too short or too long relative to the source, we compute the paragraph length ratio (output_length/source_length) and tag as `to_reject` outputs with a ratio below 0.5 or above 1.5 (i.e. a change of more than 50%).

For isolation of text complexity and to keep both datasets perfectly aligned at the example level, all examples tagged as `to_skip` or `to_reject` are **removed from both datasets**.

D Examples from Human-written and Simplified Data by Semantic Similarity

As shown in Table 2, 79% of datasets have semantic similarity of greater than 80%. We show examples here of texts with varying semantic similarity scores with their corresponding ROUGE-2 scores. Examples of sematic similarity > 0.8:

```
SEMANTIC SIMILARITY: 0.90, ROUGE-2: 0.27;
       fwedu_hw:important officials and well
            known persons who visited the islands
             wrote
       fwedu_simp:important visitors to the
            islands wrote
SEMANTIC SIMILARITY: 0.95, ROUGE-2: 0.41;
       fwedu_hw:Also, the authors now expect to
            apply their approach to other regions
            . They have a lot of work to do.
            After all, arid landscapes occupy
            about 65 million square kilometers of
             the earth's surface (this is almost
            four areas of Russia).
       fwedu_simp:The authors now plan to use
            their method in other areas. They
            have a lot of work ahead of them.
            Arid landscapes cover almost 65
            million square kilometers of the
            Earth's surface, which is roughly
            four times the size of Russia.
SEMANTIC SIMILARITY: 0.84, ROUGE-2: 0.24;
       fwedu_hw:Users frequently ask "how big
            should I make the pagefile?" There is
             no single answer to this question
            because it depends on the amount of
            installed RAM and on how much virtual
             memory that workload requires. If
            there is no other information
            available, the typical recommendation
             of 1.5 times the installed RAM is a
            good starting point. On server
            systems, you typically want to have
            sufficient RAM so that there is never
             a shortage and so that the pagefile
            is basically not used. On these
            systems, it may serve no useful
            purpose to maintain a really large
            pagefile. On the other hand, if disk
            space is plentiful, maintaining a
            large pagefile (for example, 1.5
            times the installed RAM) does not
            cause a problem, and this also
            eliminates the need to worry over how
             large to make it.
       fwedu_simp:The size of the pagefile
            depends on how much RAM your computer
             has and how much virtual memory your
```

work requires. A good starting point is to make the pagefile 1.5 times the size of your RAM. If you have a server, you should have enough RAM so that the pagefile is not used. In this case, it might not be useful to have a large pagefile. However, if you have plenty of disk space, you can make the pagefile 1.5 times the size of your RAM without any problems .

SEMANTIC SIMILARITY: 0.90, ROUGE-2: 0.19;

fwedu_hw:On its face, the USDA's decision to have participation in the NAIS be voluntary seems to solve all of the major concerns. Small and organic farmers will be able to "opt out" of participation in the NAIS if they have objections to its methodology. [FN203]

fwedu_simp:The USDA made the NAIS voluntary. This means that small and organic farmers can choose not to participate if they don't agree with how the NAIS works.

SEMANTIC SIMILARITY: 0.96, ROUGE-2: 0.43;

fwedu_hw:The ICD-11 includes a revised definition for alcohol use disorders (AUDs) and, more specifically, for alcohol dependence and the "harmful patterns of alcohol use."

fwedu_simp:The ICD-11 has changed how it defines alcohol use disorders (AUDs). It now includes a new definition for alcohol dependence and for when alcohol use causes harm.

SEMANTIC SIMILARITY: 0.95, ROUGE-2: 0.75;

fwedu_hw:Feel free to check out more of this website. Our goal is to provide rebuttals to the bad science behind young earth creationism, and honor God by properly presenting His creation.

fwedu_simp:Our goal is to provide rebuttals to the bad science behind young earth creationism, and honor God by properly presenting His creation. You can find more information on this website.

SEMANTIC SIMILARITY: 0.82, ROUGE-2: 0.50;

fwedu_hw:separate trees you simply set the CODEBASE attributes of each applet

fwedu_simp:set the CODEBASE attribute of each applet

SEMANTIC SIMILARITY: 0.93, ROUGE-2: 0.31;

fwedu_hw:Using samples taken from young and old horses, which have similar tendon properties to those of humans, the researchers performed a range of tests to profile the types, quantities, and proportions of proteins present in the tendon. Ultimately, the team found marked differences in the proteins in young and old horses.

fwedu_simp:The researchers took samples from young and old horses. They wanted to know about the proteins in the tendons of these horses. Tendons are similar in humans. The team did a series of tests to see what proteins were in the tendons. They found that young and old horses have different proteins in their tendons.

SEMANTIC SIMILARITY: 0.90, ROUGE-2: 0.80;

fwedu_hw:- Painful muscle cramps, spasms or pain in the abdomen, arms and legs

fwedu_simp:- Muscle cramps, spasms, or pain in the abdomen, arms, and legs can be very painful.

SEMANTIC SIMILARITY: 0.84, ROUGE-2: 0.13;

fwedu_hw:"We also felt it important to highlight where the use of the sea, such as bottom material extraction, aquaculture or wind energy, can be allowed," says Virtanen.

fwedu_simp:"We also think it's important to mention where we can use the sea in a good way, like taking sand or gravel from the bottom, farming fish, or making energy from wind," says Virtanen.

SEMANTIC SIMILARITY: 0.94, ROUGE-2: 0.49;

fwedu_hw:ThinkProgress noted that many states have refused to expand Medicaid coverage offered through the federal Affordable Care Act, thus preventing around 1.2 million Americans from receiving mental health care, according to the National Alliance on Mental Health.

fwedu_simp:ThinkProgress said that many states have not expanded Medicaid, which is a part of the Affordable Care Act. This means about 1.2 million Americans cannot get mental health care, according to the National Alliance on Mental Health.

SEMANTIC SIMILARITY: 0.97, ROUGE-2: 0.61;

fwedu_hw:latent heat -- the energy needed to melt solids or boil liquids. This energy is somewhat similar to chemical energy, since it is the energy associated with breaking or making molecular bonds, rather than atomic bonds. The total energy (heat) needed to melt a solid or boil a liquid is just Q = mL, where m is the mass of the liquid or solid, and L is the latent heat factor (given in either J or cal per kg) of melting or boiling.

fwedu_simp:latent heat is the energy needed to melt solids or boil liquids . This energy is similar to chemical energy because it involves breaking or making molecular bonds. The total energy needed to melt a solid or boil a liquid is calculated by multiplying the mass of the liquid or solid by the latent heat factor.

SEMANTIC SIMILARITY: 0.93, ROUGE-2: 0.51;

fwedu_hw:The glare of publicity that swirled about Yellow Thunder Camp last September when the government ordered its occupants to leave their chosen spot has faded like the leaves of autumn. The traditional but transient tepees have been

supplemented with a geodesic dome. The legal battle which will determine the camp's future drags on in nearby Rapid City.
fwedu_simp:The glare of publicity that swirled around Yellow Thunder Camp last September when the government ordered its occupants to leave their chosen spot has faded. The campers have added a new, dome-shaped shelter to their traditional tepees. The legal fight about the camp's future is still going on in Rapid City.
SEMANTIC SIMILARITY: 0.98, ROUGE-2: 0.74;
fwedu_hw:The U.S. Geological Survey's National Wildlife Health Center verified the disease in a little brown bat found this month in North Bend, about 30 miles east of Seattle.
fwedu_simp:The U.S. Geological Survey's National Wildlife Health Center found a disease in a little brown bat in North Bend, which is about 30 miles east of Seattle.
SEMANTIC SIMILARITY: 0.94, ROUGE-2: 0.40;
fwedu_hw:Replacing native species with non-native species does not necessarily cause biotic homogenization. If different communities of non-native species replace native species at cities around the world then biotic differentiation rather than homogenization will have occurred. However, there is no evidence that this is happening. Many studies have shown that the extirpation of native species in urban environments and the influx and non-native invasive species is leading to global biotic homogenization. For example a study of urban bird populations from two distant locations (Ohio and California) found urban populations to be much more similar to each other than rural populations the same distance apart (19). Similarly, a study in Canada found that the ecology of cities across the country was becoming increasingly alike with many of the same species found in cities nationwide (14).
fwedu_simp:Biotic homogenization is when different areas start to have the same species. This can happen if non-native species replace native species in cities around the world. However, this is not happening. Many studies have shown that native species are disappearing from cities and non-native species are moving in. This is causing biotic homogenization worldwide. For example, a study found that urban bird populations in Ohio and California are much more alike than bird populations in rural areas that are the same distance apart. Another study in Canada found that cities across the country are becoming more alike, with many of the same species found in cities everywhere.
SEMANTIC SIMILARITY: 0.91, ROUGE-2: 0.19;
fwedu_hw:An independent panel of technical experts convened by the American Chemical Society Green Chemistry Institute formally judged the 2017 submissions from among scores of nominated technologies and made recommendations to EPA for the 2017 winners. The 2017 awards event will be held in conjunction with the 21st Annual Green Chemistry and Engineering Conference.
fwedu_simp:An independent group of experts looked at many technologies and chose the best ones for the 2017 awards. They recommended these winners to the EPA. The 2017 awards ceremony will be held at the same time as a conference on green chemistry.
SEMANTIC SIMILARITY: 0.94, ROUGE-2: 0.38;
fwedu_hw:Only $24.00 and a pair of high boots was all it took for the first property owner to purchase the land where the now renowned Pioneer Courthouse Square is located. The block was the site for Portland's first school. Shortly thereafter, it became the Portland Hotel where it served as a social center. The hotel was demolished in 1951 to make room for the automobile with installation of a full city block of parking. Due to progressive civic leadership in the 1970's, Portland worked to revitalize its downtown, including a move away from the use of automobiles and back toward mass transit. The demolition of the parking garage and creation of Pioneer Courthouse Square remains a major landmark of this effort.
fwedu_simp:Only $24.00 and a pair of boots was all it took for the first person to buy the land where Pioneer Courthouse Square is now. This block was once home to Portland's first school. Later, it became the Portland Hotel, where people would meet and socialize. The hotel was torn down in 1951 to make room for cars. In the 1970s, Portland's leaders decided to make the city more people-friendly. They wanted to reduce the use of cars and increase the use of public transportation. As part of this effort, the parking garage was removed, and Pioneer Courthouse Square was created.
SEMANTIC SIMILARITY: 0.95, ROUGE-2: 0.39;
fwedu_hw:The wearing of gowns at formals is compulsory at some colleges and various other traditions are usually observed, including grace said in Latin or English. The wearing of gowns may sometimes constitute the only dress code; in other cases, formal wear (for example, a lounge

```
            suit for men or equivalent for women)
             is required in addition to, or
            instead of, the gown.
       fwedu_simp:The wearing of gowns at
            formals is required at some colleges
            and some other traditions are
            followed, like saying grace in Latin
            or English. In some places, wearing a
             gown is the only dress code, while
            in others, you also need to wear
            formal clothes (like a suit for men
            or something similar for women) along
             with the gown.
```

Examples of sematic similarity < 0.5:

```
SEMANTIC SIMILARITY: 0.42, ROUGE-2: 0.04;
       fwedu_hw:and wife, between teacher and
            pupil, between friend and friend,
            between employer and employee, and
            between religious teacher and
            disciple. It no longer means honoring
             the gods which inhabit the six
            different directions.
       fwedu_simp:The concept of "reverence" has
             changed over time. It used to mean
            showing respect to gods in different
            directions.
SEMANTIC SIMILARITY: 0.09, ROUGE-2: 0.00;
       fwedu_hw:- Press Ctrl + 2 to add more
            text boxes. Press Ctrl + shift + 2 to
             adjust text box.
       fwedu_simp:(Note: Please provide your
            output in the format specified above,
             ensuring it is free of grammatical
            errors and easy to read.)
SEMANTIC SIMILARITY: 0.38, ROUGE-2: 0.00;
       fwedu_hw:his bark is worse than his bite,
             he is bad-tempered but harmless
       fwedu_simp:This person is grumpy, but he
            won't hurt you.
SEMANTIC SIMILARITY: 0.44, ROUGE-2: 0.00;
       fwedu_hw:said to have sworn, under duress
            , that he
       fwedu_simp:The person was forced to say
            something, but he didn't really mean
            it.
SEMANTIC SIMILARITY: 0.42, ROUGE-2: 0.18;
       fwedu_hw:Woolworth's sit-in-a photograph
            that has become the image used in
            history books and
       fwedu_simp:Woolworth's sit-in was a
            protest where African American
            students sat at a lunch counter to
            demand equal rights.
SEMANTIC SIMILARITY: 0.48, ROUGE-2: 0.35;
       fwedu_hw:Sadly, as Hentoff points out,
            the American Civil Liberties Union
            has remained silent on this latest
            gross violation of our rights. So,
            too, has most of the fifth estate.
       fwedu_simp:The American Civil Liberties
            Union has not spoken out against this
             big mistake of our rights. Most of
            the news media has also been quiet.
SEMANTIC SIMILARITY: 0.35, ROUGE-2: 0.24;
       fwedu_hw:and operated at 33 MHz and 20
            MIPS. ...Many thanks to Robert B
            Garner - who
       fwedu_simp:The computer was made by Intel
             and operated at 33 million cycles
            per second and 20 million
            instructions per second.
SEMANTIC SIMILARITY: 0.41, ROUGE-2: 0.03;
       fwedu_hw:3) Low investment costs of about
             ~900Euro and a simple construction
            with locally available materials. (
            Costs were about 300 Euro in Kenya in
             2004, but now with raising steel
            prices costs increased...) .
       fwedu_simp:The cost of building a simple
            water filter is relatively low. It
            costs about 900 Euro. The
            construction is also simple and can
            be made using materials that are
            easily available locally.
SEMANTIC SIMILARITY: 0.48, ROUGE-2: 0.32;
       fwedu_hw:you are near the surface of the
            Earth, regardless of what the object
            is
       fwedu_simp:The surface of the Earth is
            the outermost solid layer of our
            planet.
SEMANTIC SIMILARITY: 0.36, ROUGE-2: 0.09;
       fwedu_hw:upon his visage, rather than
            pure devotion, such as one might
       fwedu_simp:The person's face showed more
            of a sense of duty than pure love.
SEMANTIC SIMILARITY: 0.14, ROUGE-2: 0.00;
       fwedu_hw:- Genetic screens in human cells
             using the CRISPR-Cas9 system.
            Science 343, 80-84 (2014) , , &
       fwedu_simp:Simplification of the text
            should be provided in the format
            specified above.
SEMANTIC SIMILARITY: 0.11, ROUGE-2: 0.00;
       fwedu_hw:Strategies you implement are
            usually defined as the tone of your
            information. Here is the summary of
            tone types:
       fwedu_simp:(Note: Please provide your
            output in the format specified above,
             ensuring it is clear, well-organized
            , and free of grammatical errors.)
SEMANTIC SIMILARITY: 0.08, ROUGE-2: 0.00;
       fwedu_hw:- Mathematics - Knowledge of
            arithmetic, algebra, geometry,
            calculus, statistics, and their
            applications.
       fwedu_simp:Simplification of the text
            should be done in the same format as
            the examples provided.
SEMANTIC SIMILARITY: 0.14, ROUGE-2: 0.00;
       fwedu_hw:Art. 304, consists of two
            clauses, and each clause operates as
            a proviso to Arts. 301 and 303.
       fwedu_simp:The law has two parts. Each
            part is connected to other laws.
SEMANTIC SIMILARITY: 0.17, ROUGE-2: 0.00;
       fwedu_hw:See also: What is the meaning of
             Jurisdiction, Lawyer, Court, Law,
            State?
       fwedu_simp:Simplification of the text
            goes here.
SEMANTIC SIMILARITY: 0.43, ROUGE-2: 0.00;
       fwedu_hw:as an art form, let alone to
            caricature.34 De Bruycker is to Ghent
             perhaps what the
       fwedu_simp:Art is a way of expressing
            oneself, but it's not just about
```

```
            making something look funny or
            different.
SEMANTIC SIMILARITY: 0.44, ROUGE-2: 0.12;
       fwedu_hw:Figure 5. Mass fluctuations and
            collapse thresholds in cold dark
            matter models. The horizontal dotted
            lines show the value of the
            extrapolated collapse overdensity
            crit(z) at the indicated redshifts.
            Also shown is the value of (M) for
            the cosmological parameters given in
            the text (solid curve), as well as (M
            ) for a power spectrum with a cutoff
            below a mass M = 1.7 x 108 M (short-
            dashed curve), or M = 1.7 x 1011 M (
            long-dashed curve). The intersection
            of the horizontal lines with the
            other curves indicate, at each
            redshift z, the mass scale (for each
            model) at which a 1 - fluctuation is
            just collapsing at z (see the
            discussion in the text).
       fwedu_simp:The diagram shows how the mass
             of objects in the universe changes
            over time. The horizontal lines show
            the point at which a group of objects
             would start to collapse under their
            own gravity. The solid line shows how
             the mass of objects changes in the
            universe based on the given
            information. The other lines show how
             the mass of objects would change if
            there were a limit to the size of
            objects in the universe. The points
            where the horizontal lines intersect
            with the other lines show the mass of
             objects at each point in time that
            would just start to collapse.
SEMANTIC SIMILARITY: 0.45, ROUGE-2: 0.00;
       fwedu_hw:- Can you think of other cases
            where a government has addressed its
            previous wrongdoing?
       fwedu_simp:- Yes, there are several
            examples.
SEMANTIC SIMILARITY: 0.06, ROUGE-2: 0.00;
       fwedu_hw:- Wages - A comparison of wages
            between men and women, children and
            adults.
       fwedu_simp:Simplification of the text
            should be provided in the format
            specified above.
```

E Examples from Human-written and Simplified Data by ROUGE-2

In Table 2, we used ROUGE-2 (R2) thresholds to define the level of lexical overlap.

Examples of low lexical overlap ($0 < R2 \leq 0.4$):

```
ROUGE-2: 0.19;
       fwedu_hw:An independent panel of
            technical experts convened by the
            American Chemical Society Green
            Chemistry Institute formally judged
            the 2017 submissions from among
            scores of nominated technologies and
            made recommendations to EPA for the
            2017 winners. The 2017 awards event
            will be held in conjunction with the
            21st Annual Green Chemistry and
            Engineering Conference.
       fwedu_simp:An independent group of
            experts looked at many technologies
            and chose the best ones for the 2017
            awards. They recommended these
            winners to the EPA. The 2017 awards
            ceremony will be held at the same
            time as a conference on green
            chemistry.
ROUGE-2: 0.38;
       fwedu_hw:Only $24.00 and a pair of high
            boots was all it took for the first
            property owner to purchase the land
            where the now renowned Pioneer
            Courthouse Square is located. The
            block was the site for Portland's
            first school. Shortly thereafter, it
            became the Portland Hotel where it
            served as a social center. The hotel
            was demolished in 1951 to make room
            for the automobile with installation
            of a full city block of parking. Due
            to progressive civic leadership in
            the 1970's, Portland worked to
            revitalize its downtown, including a
            move away from the use of automobiles
             and back toward mass transit. The
            demolition of the parking garage and
            creation of Pioneer Courthouse Square
             remains a major landmark of this
            effort.
       fwedu_simp:Only $24.00 and a pair of
            boots was all it took for the first
            person to buy the land where Pioneer
            Courthouse Square is now. This block
            was once home to Portland's first
            school. Later, it became the Portland
             Hotel, where people would meet and
            socialize. The hotel was torn down in
             1951 to make room for cars. In the
            1970s, Portland's leaders decided to
            make the city more people-friendly.
            They wanted to reduce the use of cars
             and increase the use of public
            transportation. As part of this
            effort, the parking garage was
            removed, and Pioneer Courthouse
            Square was created.
ROUGE-2: 0.10;
       fwedu_hw:- 2002 - 2011 is the ten years
            preceding the ratings evaluation, and
       fwedu_simp:- 2002 to 2011 was the time
            before the ratings were checked.
ROUGE-2: 0.39;
       fwedu_hw:The wearing of gowns at formals
            is compulsory at some colleges and
            various other traditions are usually
            observed, including grace said in
            Latin or English. The wearing of
            gowns may sometimes constitute the
            only dress code; in other cases,
            formal wear (for example, a lounge
            suit for men or equivalent for women)
             is required in addition to, or
            instead of, the gown.
       fwedu_simp:The wearing of gowns at
            formals is required at some colleges
            and some other traditions are
```

```
            followed, like saying grace in Latin
            or English. In some places, wearing a
             gown is the only dress code, while
            in others, you also need to wear
            formal clothes (like a suit for men
            or something similar for women) along
             with the gown.
```

Examples of medium lexical overlap ($0.4 < R2 \leq 0.8$**):**

```
ROUGE-2: 0.68;
       fwedu_hw:HDTV technology is estimated
            that this will be the future of
            television standards, so a senior
            researcher in the field of systems
            and management strategies Dr. Indu
            Singh predicts that the world market
            for HDTV would reach 250 billion
            dollars per year (year 2010).
       fwedu_simp:HDTV technology is expected to
             be the future of television
            standards. Dr. Indu Singh, a senior
            researcher in the field of systems
            and management strategies, predicts
            that the world market for HDTV will
            reach $250 billion per year by 2010.
ROUGE-2: 0.74;
       fwedu_hw:Prophetically, he feels the need
             to plead for ten years of life so
            that:
       fwedu_simp:Prophetically, he feels the
            need to ask for ten more years of
            life so that:
ROUGE-2: 0.47;
       fwedu_hw:Most common palm species are
            Elaeis guineensis and Borassus
            aethiopium (rhun palm).
       fwedu_simp:The two most common types of
            palm trees are Elaeis guineensis and
            Borassus aethiopium, also known as
            the rhun palm.
ROUGE-2: 0.43;
       fwedu_hw:The old man almost immediately
            fell asleep; but the boy, Minokichi,
            lay awake a long time, listening to
            the awful wind, and the continual
            slashing of the snow against the door
            . The river was roaring; and the hut
            swayed and creaked like a junk at sea
            . It was a terrible storm; and the
            air was every moment becoming colder;
             and Minokichi shivered under his
            rain-coat. But at last, in spite of
            the cold, he too fell asleep.
       fwedu_simp:The old man fell asleep right
            away. But the boy, Minokichi, lay
            awake for a long time. He listened to
             the strong wind and the snow hitting
             the door. The river was making a
            loud noise, and the hut was swaying
            and creaking like a boat at sea. It
            was a very bad storm, and the air was
             getting colder every minute.
            Minokichi was shivering under his
            raincoat. But eventually, despite the
             cold, he fell asleep too.
ROUGE-2: 0.51;
       fwedu_hw:The glare of publicity that
            swirled about Yellow Thunder Camp
            last September when the government
            ordered its occupants to leave their
            chosen spot has faded like the leaves
             of autumn. The traditional but
            transient tepees have been
            supplemented with a geodesic dome.
            The legal battle which will determine
             the camp's future drags on in nearby
             Rapid City.
       fwedu_simp:The glare of publicity that
            swirled around Yellow Thunder Camp
            last September when the government
            ordered its occupants to leave their
            chosen spot has faded. The campers
            have added a new, dome-shaped shelter
             to their traditional tepees. The
            legal fight about the camp's future
            is still going on in Rapid City.
ROUGE-2: 0.41;
       fwedu_hw:Also, the authors now expect to
            apply their approach to other regions
            . They have a lot of work to do.
            After all, arid landscapes occupy
            about 65 million square kilometers of
             the earth's surface (this is almost
            four areas of Russia).
       fwedu_simp:The authors now plan to use
            their method in other areas. They
            have a lot of work ahead of them.
            Arid landscapes cover almost 65
            million square kilometers of the
            Earth's surface, which is roughly
            four times the size of Russia.
ROUGE-2: 0.75;
       fwedu_hw:Feel free to check out more of
            this website. Our goal is to provide
            rebuttals to the bad science behind
            young earth creationism, and honor
            God by properly presenting His
            creation.
       fwedu_simp:Our goal is to provide
            rebuttals to the bad science behind
            young earth creationism, and honor
            God by properly presenting His
            creation. You can find more
            information on this website.
```

Examples of high lexical overlap ($0.8 < R2 < 1$**):**

```
ROUGE-2: 0.85;
       fwedu_hw:That same year, the FDA and EPA
            issued a recommendation that pregnant
             women and young children eat no more
             than two servings, or 12 ounces, of
            salmon and other low-mercury fish
            each week.
       fwedu_simp:The FDA and EPA suggested that
             pregnant women and young children
            eat no more than two servings, or 12
            ounces, of salmon and other low-
            mercury fish each week.
ROUGE-2: 0.84;
       fwedu_hw:With a little imagination, other
             services could be provided as well.
       fwedu_simp:With a little imagination,
            other services could be provided too.
ROUGE-2: 0.82;
       fwedu_hw:o Suggests questions to help
            facilitate professional development
```

```
            group discussions, especially among
            peers
       fwedu_simp:o Suggests questions to help
            facilitate group discussions,
            especially among peers
ROUGE-2: 0.90;
       fwedu_hw:tendonitis. The flattened arch
            pulls on calf muscles and keeps the
            Achilles tendon under tight strain.
            This constant mechanical stress on
            the heel and tendon can cause
            inflammation, pain and swelling
       fwedu_simp:tendonitis. The flattened arch
             pulls on calf muscles and keeps the
            Achilles tendon under tight strain.
            This constant stress on the heel and
            tendon can cause pain and swelling.
```

Examples of exact match ($R2 = 1$):

```
ROUGE-2: 1.00;
       fwedu_hw:- Does the modal not show a
            coupon code? Then you can click
            directly in the big blue button "
            VISIT Hidden24 VPN
       fwedu_simp:- Does the modal not show a
            coupon code? Then you can click
            directly in the big blue button "
            VISIT Hidden24 VPN"
ROUGE-2: 1.00;
       fwedu_hw:- IVF through implanting
            multiple embryos can be one way of
            getting science to help with the
            process
       fwedu_simp:IVF through implanting
            multiple embryos can be one way of
            getting science to help with the
            process.
ROUGE-2: 1.00;
       fwedu_hw:For more information about the
            program contact Stoughton at
            435-259-7985 or email email@example.
            com.
       fwedu_simp:For more information about the
             program, contact Stoughton at
            435-259-7985 or email email@example.
            com.
ROUGE-2: 1.00;
       fwedu_hw:An earthworm's home, and the
            dirt around it, can be called a
            factory. This factory makes a special
             kind of dirt called topsoil.
       fwedu_simp:An earthworm's home and the
            dirt around it can be called a
            factory. This factory makes a special
             kind of dirt called topsoil.
ROUGE-2: 1.00;
       fwedu_hw:Tim Wilson will be speaking to
            The New Zealand Initiative in:
       fwedu_simp:Tim Wilson will be speaking to
             The New Zealand Initiative in:
ROUGE-2: 1.00;
       fwedu_hw:- extending far in width; broad:
             deep lace; a deep border.
       fwedu_simp:- extending far in width;
            broad: deep lace; a deep border.
```

Examples of exact mismatch ($R2 = 0$):

```
ROUGE-2: 0.00;
       fwedu_hw:ensure that every medical issue
            receives attention.
       fwedu_simp:Medical issues should get
            attention.
ROUGE-2: 0.00;
       fwedu_hw:- Press Ctrl + 2 to add more
            text boxes. Press Ctrl + shift + 2 to
             adjust text box.
       fwedu_simp:(Note: Please provide your
            output in the format specified above,
             ensuring it is free of grammatical
            errors and easy to read.)
ROUGE-2: 0.00;
       fwedu_hw:judicial decorum when expressing
             himself on conservation matters. . .
             ."
       fwedu_simp:The judge spoke about
            conservation in a respectful and
            proper way.
ROUGE-2: 0.00;
       fwedu_hw:his bark is worse than his bite,
             he is bad-tempered but harmless
       fwedu_simp:This person is grumpy, but he
            won't hurt you.
ROUGE-2: 0.00;
       fwedu_hw:*An earlier version of this
            article misstated the study's
            benchmark for deficit reduction.
       fwedu_simp:The article previously
            mentioned the wrong target for
            reducing the deficit.
ROUGE-2: 0.00;
       fwedu_hw:said to have sworn, under duress
            , that he
       fwedu_simp:The person was forced to say
            something, but he didn't really mean
            it.
ROUGE-2: 0.00;
       fwedu_hw:and resulted in considerable
            damage.
       fwedu_simp:The hurricane caused a lot of
            damage.
ROUGE-2: 0.00;
       fwedu_hw:- Thomas, B. 2009. Did Humans
            Evolve from 'Ardi'? Acts & Facts. 38
            (11): 8-9.
       fwedu_simp:Simplified Text:
"Thomas wrote about a discovery called 'Ardi' in
     2009. He asked if humans evolved from this
    ancient creature.
ROUGE-2: 0.00;
       fwedu_hw:Strategies you implement are
            usually defined as the tone of your
            information. Here is the summary of
            tone types:
       fwedu_simp:(Note: Please provide your
            output in the format specified above,
             ensuring it is clear, well-organized
            , and free of grammatical error
```

F Outliers

To improve visualizations, we clipped outliers (Flesch Reading Ease) which only accounts for 3.85% (fwedu_hw) and 1.25% (fwedu_simp), and also removed outliers (Sentence Split Difference, Compression Level, Dependency Tree Depth Ra-

tio) which only accounts 2.91% as a whole. Total examples for each dataset is 26,315,220. This section defines, quantifies, and illustrates the outliers.

F.1 Outliers: Flesch Reading Ease

Flesch Reading Ease (FRE) is interpreted as 0 to 100 but the FRE formula does not enforce boundaries, for this reason we clip negative values to 0 and clip to 100 if FRE is beyond 100. Negative FRE values can happen for dense paragraphs with very long sentences (typically, complex sentences) with long words. While FRE of greater than 100 can happen for paragraphs with very short sentences with short words. The percentage of outliers are as follows: 3.85% for `fwedu_hw` and 1.25% for `fwedu_simp` examples.

Examples of outliers are provided below.

```
# fwedu_hw
FRE: 100.00; "Come out of her, my people, lest
     you take part of her sins, lest you share in
FRE: 112.09; - Press Ctrl + 2 to add more text
     boxes. Press Ctrl + shift + 2 to adjust text
      box.
FRE: 102.53; Do you know the name of the bird
     group you are looking for?

# fwedu_simp
FRE: 103.01; - 2002 to 2011 was the time before
     the ratings were checked.
FRE: 103.70; - As these experts say, we need to
     start
FRE: 103.65; The eastern part of the bridge
     weighs over 3,800 tons. The western part
     weighs over 1,000 tons.

#fwedu_hw
FRE: -15.65; Zambia started its accelerated
     malaria control campaign in 2003 when
     approximately 500,000 insecticide-treated
     nets were distributed and artemisinin-based
     combination therapy (ACT) started in seven
     pilot districts through a grant from the UN-
     backed Global Fund to fight AIDS,
     Tuberculosis and Malaria.
FRE: -11.91; NASA Image: ISS015E13648 - View of
     Expedition 15 astronaut and Flight Engineer,
      Clayton Anderson, working with test samples
      in the Human Research Facility - 2
     Refrigerated Centrifuge for the Nutritional
     Status Assessment experiment to help
     understand human physiologic changes during
     long-duration space flight.
FRE: -1.59; o Suggests questions to help
     facilitate professional development group
     discussions, especially among peers

# fwedu_simp
FRE: -53.65230769230766; Interconnectedness,
     empowerment, cooperation, relationships,
     partnership, flexibility, and diversity are
     key to realizing opportunities and creating
     sustainable systems. This includes nations,
     organizations, and communities working
     together effectively.
FRE: -18.44999999999996; Environmental engineers
      with experience in project management,
     regulatory compliance, environmental
     compliance, and engineering design tend to
     earn more, according to data from PayScale
     (2017).
FRE: -8.098461538461521; Occupational therapists
      help people do everyday activities by
     giving them exercises and practice.
```

F.2 Outliers: Sentence Split Difference, Compression Level, Dependency Tree Depth Ratio

For these metrics, we identified outliers by computing the interquartile range (IQR). We compute bounds as $lower\_bound = Q1 - 3 * IQR$ and $upper\_bound = Q3 + 3 * IQR$, where $IQR = Q3 - Q1$ and Q1 and Q3 stands for Quartile 1 and 3, respectively. Usually, 1.5 was used to compute the bounds but we increased it to 3 to widen the threshold and make the tagging of outliers less aggressive. The percentage for each outlier type are as follows: sentence split difference (0.93%), compression level (0.42%), dependency tree depth ratio (1.74%). Combined and without duplicates, it accounts for 2.91% of data. **We removed these outliers for the visualization** in Figure 2. We give examples of outliers below.

Example of Compression Level outliers:

```
Compression level: 1.84;
       fwedu_hw:Reproduction - Ovoviviparous.
            Embryos are ovophagous; smaller
            siblings are consumed by larger
            siblings. Litters range from 4 to 6
            pups, and size at birth is 1.9 to 2.6
             ft [60 to 80 cm].
       fwedu_simp:Reproduction - Ovoviviparous.
            This means that the mother gives
            birth to live young, but they develop
             inside eggs. The babies eat the eggs
             they hatch from, and if there are
            smaller siblings, the bigger ones
            might eat them too. The litter size
            is usually between 4 and 6 pups, and
            they are born after about 1.9 to 2.6
            feet [60 to 80 cm] long.
Compression level: 1.80;
       fwedu_hw:- Centers for Disease Control
            and Prevention. Update: respiratory
            syncytial virus activity - United
            States, 1998-1999 Season. MMWR Morb
            Mortal Wkly Rep. 1999;48:1104-15.
       fwedu_simp:Simplified Text:
"The Centers for Disease Control and Prevention
    (CDC) reported on the respiratory syncytial
    virus (RSV) activity in the United States
    for the 1998-1999 season. RSV is a virus
    that affects the respiratory system. The CDC
     monitored RSV activity and reported the
    findings in a weekly report.
Compression level: 2.48;
```

```
       fwedu_hw:Koh, L.P. and Wilcove, D.S.
           (2008) 'Is oil palm agriculture
           really dest
       fwedu_simp:Koh and Wilcove studied the
           impact of oil palm agriculture on the
            environment. They found that oil
           palm plantations can lead to
           deforestation and habitat loss for
           many species.
Compression level: 2.12;
       fwedu_hw:LEEDS, A. Introduction. In:
           WILBERT, J. (Ed.). The evolution of
           horticultural systems in native south
            America: causes and consequences - A
            Symposium. Caracas: Sociedad de
           Ciencias Naturales La Salle, 1961. p.
            1-12. [ Links ]
       fwedu_simp:The evolution of horticultural
            systems in native South America is a
            topic of great interest. This
           symposium aims to explore the causes
           and consequences of these changes.
           The book you are reading is a
           collection of papers presented at
           this symposium. It covers various
           aspects of horticulture in South
           America, from the early days to the
           present. The authors of these papers
           are experts in their fields and have
           contributed significantly to our
           understanding of this subject.
Compression level: 1.81;
       fwedu_hw:of the legion to carry out
           special duties. Marius thus created a
            fully
       fwedu_simp:Marius created a special group
            of soldiers within the Roman legion.
            This group was responsible for
           carrying out specific tasks.
```

Example of Dependency Tree Depth Ratio outliers:

```
Max Dependency Tree Depth Ratio: 1.83;
       fwedu_hw:The wearing of gowns at formals
           is compulsory at some colleges and
           various other traditions are usually
           observed, including grace said in
           Latin or English. The wearing of
           gowns may sometimes constitute the
           only dress code; in other cases,
           formal wear (for example, a lounge
           suit for men or equivalent for women)
            is required in addition to, or
           instead of, the gown.
       fwedu_simp:The wearing of gowns at
           formals is required at some colleges
           and some other traditions are
           followed, like saying grace in Latin
           or English. In some places, wearing a
            gown is the only dress code, while
           in others, you also need to wear
           formal clothes (like a suit for men
           or something similar for women) along
            with the gown.
Max Dependency Tree Depth Ratio: 2.33;
       fwedu_hw:- Press Ctrl + 2 to add more
           text boxes. Press Ctrl + shift + 2 to
            adjust text box.
       fwedu_simp:(Note: Please provide your
           output in the format specified above,
            ensuring it is free of grammatical
           errors and easy to read.)
Max Dependency Tree Depth Ratio: 2.00;
       fwedu_hw:Reade, Julian. Assyrian
           Sculpture. London: The British Museum
           ; and Cambridge, MA: Harvard
           University Press, 1983, repr. 1994.
       fwedu_simp:Julian Reade wrote a book
           about Assyrian sculpture. It was
           published by the British Museum in
           London and Harvard University Press
           in Cambridge, MA. The book was first
           published in 1983 and then again in
           1994.
Max Dependency Tree Depth Ratio: 1.80;
       fwedu_hw:- Centers for Disease Control
           and Prevention. Update: respiratory
           syncytial virus activity - United
           States, 1998-1999 Season. MMWR Morb
           Mortal Wkly Rep. 1999;48:1104-15.
       fwedu_simp:Simplified Text:
"The Centers for Disease Control and Prevention
    (CDC) reported on the respiratory syncytial
    virus (RSV) activity in the United States
    for the 1998-1999 season. RSV is a virus
    that affects the respiratory system. The CDC
     monitored RSV activity and reported the
    findings in a weekly report.
Max Dependency Tree Depth Ratio: 2.00;
       fwedu_hw:Clarke disclosed no relevant
           relationships with industry. Co-
           authors disclosed multiple relevant
           relationships with industry.
       fwedu_simp:Clarke did not have any
           relationships with companies that
           could affect the study. The other
           authors had relationships with
           companies that could affect the study
           .
```

Example of Sentence Split Difference outliers:

```
sentence_splits_diff: 54.00;
       fwedu_hw:1 cross-section anisotropically
            etched groove in (100) silicon 2
            schematic of reflux reactor 3
            enlarged section of full-mask as laid
             out on L-edit 4 full mask as laid
            out on L-edit 5 enlarged section of
            full-mask showing large springs as
            laid out on L-edit 6 grid which lines
             up with circles on alignment mask as
             laid out on L-edit 7 alignment mask
            as laid out on L-edit 8 enlarged
            section of full mask showing
            diaphragms as laid out on L-edit 9
            two lines , one is obviously more
            undercut than the other (mic.) 10
            feature 'lines' after 40 minutes of
            etching in ecolite (SEM) 11 blown-up
            view of feature 'lines' after 40 min.
             etching in ecolite (SEM) 12 200\B5\m
             diameter circle etched out to form
            square (mic.) 13 200\B5\m diameter
            circle which has been etched out to
            form square (SEM) 14 400\B5\m
            diameter circle which has been etched
             out to form square (SEM) 15 silicon
            dioxide capillary bridges (mic.) 16
            square oxide feature etched away at
            its convex corners (SEM) 17 square
            oxide feature etched away at its
            convex corners (SEM) 18 cluster of
            etch holes in the oxide layer on back
             side of the wafer (SEM) 19 pinholes
            in boron-doped silicon channel under
            capillaries (mic.) 20 silicon channel
             etched under a broken-off cantilever
             (mic.) 21 pitted silicon surface of
            an incompletely -etched diaphragm (
            SEM) 22 square of boron-doped Si
            formed by etching a trough around it
            (SEM) 23 blown-up view of the boron-
            doped silicon square (SEM) 24
            diaphragm etched through to boron-
            doped rough side of wafer (mic.) 25
            boron-doped silicon capillary bridges
             (SEM) 26 blown-up view of free boron
            -doped silicon bridge (SEM) 27
            silicon dioxide capillary bridges (
            SEM) 28 free boron-doped silicon
            little spring (SEM) 29 free silicon-
            dioxide cantilever (SEM) 30 boron-
            doped silicon cantilevers showing
            (111) crystal planes (SEM) 31 smooth
            (polished side of wafer) boron-doped
            silicon diaphragm (SEM)Literature
            Cited
       fwedu_simp:1. A groove was cut into a
            silicon wafer in a special way so
            that it would be etched differently
            in different directions.
2. This is a picture of a special machine used
    to make chemicals mix together in a certain
    way.
3. This is a close-up of the full design for
    making the silicon wafer, as seen on a
    computer screen.
4. This is the full design for making the
    silicon wafer, as seen on a computer screen.
5. This is a close-up of the full design for
    making the silicon wafer, showing the big
    springs, as seen on a computer screen.
6. This is a grid that lines up with circles on
    another design, as seen on a computer screen
    .
7. This is the design for lining up the circles,
     as seen on a computer screen.
8. This is a close-up of the full design for
    making the silicon wafer, showing the
    diaphragms, as seen on a computer screen.
9. These are two lines, one of which is more
    worn away than the other.
10. This is a picture of lines that were etched
    into the silicon wafer after 40 minutes of
    etching.
11. This is a close-up of the lines that were
    etched into the silicon wafer after 40
    minutes of etching.
12. This is a circle that was etched out to form
     a square.
13. This is a picture of a circle that was
    etched out to form a square.
14. This is a picture of a circle that was
    etched out to form a square.
15. These are small bridges made of silicon
    dioxide.
16. This is a square feature that was etched
    away at its corners.
17. This is a picture of a square feature that
    was etched away at its corners.
18. These are small holes in the oxide layer on
    the back of the wafer.
19. These are small holes in the silicon channel
     under the capillaries.
20. This is a silicon channel that was etched
    under a broken-off cantilever.
21. This is a pitted silicon surface of a
    diaphragm that was not fully etched.
22. This is a square of boron-doped silicon that
     was formed by etching a trough around it.
23. This is a close-up of the boron-doped
    silicon square.
24. This is a diaphragm that was etched through
    to the rough side of the wafer.
25. These are small bridges made of boron-doped
    silicon.
26. This is a close-up of a free boron-doped
    silicon bridge.
27. These are small bridges made of silicon
    dioxide.
28. This is a small, free boron-doped silicon
    spring.
29. This is a small, free silicon dioxide
    cantilever.
30. These are boron-doped silicon cantilevers
    that show the crystal planes.
31. This is a smooth, polished side of a boron-
    doped silicon diaphragm.
sentence_splits_diff: 55.00;
       fwedu_hw:55 new units: Huscarls (Spear),
            Huscarls (Axe), Mounted Huscarls,
            Berserkers, Well-Equipped Shieldwall
            (Offensive), Shieldwall (Offensive),
            Hirdsmen, Dismounted Hirdsmen, Picked
             Irish Foot (Axe), Irish Foot (Axe),
            Irish Kerns, (Dark Age) Armoured
            Lancers, Dismounted Armoured Lancers,
             (Dark Age) Lancers, Dismounted
            Lancers, (Dark Age) Armoured Cavalry
```

```
            (Light Spear), Dismounted Armoured
            Cavalry, (Dark Age) Cavalry (Light
            Spear), Dismounted Cavalry,
            Crossbowmen, Light Crossbowmen,
            Byzantine Kataphraktoi, Tagmatic
            Lancers & Archers, Thematic Lancers &
             Archers, Varangian Guard (Early),
            Byzantine Skutatoi, Byzantine Raw
            Skutatoi, Byzantine Massed Archers,
            Byzantine Light Archers, Byzantine
            Skutatoi & Archers, Raw Byzantine
            Skutatoi & Archers, Light Horse
            Archers (Pecheneg/Cuman), Horse
            Archers (Pecheneg/Cuman), Muslim
            Spearmen, Raw Muslim Spearmen,
            Veteran Muslim Spearmen, 'Abid al-
            shira, Muslim Irregular Foot,
            Armoured Muslim Lancers (Superior),
            Dismounted (Superior) Armoured Muslim
             Lancers, Armoured Muslim Lancers (
            Average), Dismounted (Average)
            Armoured Muslim Lancers, Muslim
            Lancers, Ghilman, Dismounted Ghilman,
             Muslim Cavalry (Light Spear), Muslim
             War Elephants, Muslim Light Horse (
            Javelins), Muslim Light Foot Archers,
             Muslim Light Javelinmen, Naffatun,
            Veteran Dailami Foot, Dailami Foot,
            Superior Indian Lancers, Indian
            Lancers.
       fwedu_simp:1. Huscarls (Spear)
2. Huscarls (Axe)
3. Mounted Huscarls
4. Berserkers
5. Well-Equipped Shieldwall (Offensive)
6. Shieldwall (Offensive)
7. Hirdsmen
8. Dismounted Hirdsmen
9. Picked Irish Foot (Axe)
10. Irish Foot (Axe)
11. Irish Kerns
12. (Dark Age) Armoured Lancers
13. Dismounted Armoured Lancers
14. (Dark Age) Lancers
15. Dismounted Lancers
16. (Dark Age) Armoured Cavalry (Light Spear)
17. Dismounted Armoured Cavalry
18. (Dark Age) Cavalry (Light Spear)
19. Dismounted Cavalry
20. Crossbowmen
21. Light Crossbowmen
22. Byzantine Kataphraktoi
23. Tagmatic Lancers & Archers
24. Thematic Lancers & Archers
25. Varangian Guard (Early)
26. Byzantine Skutatoi
27. Byzantine Raw Skutatoi
28. Byzantine Massed Archers
29. Byzantine Light Archers
30. Byzantine Skutatoi & Archers
31. Raw Byzantine Skutatoi & Archers
32. Light Horse Archers (Pecheneg/Cuman)
33. Horse Archers (Pecheneg/Cuman)
34. Muslim Spearmen
35. Raw Muslim Spearmen
36. Veteran Muslim Spearmen
37. 'Abid al-shira
38. Muslim Irregular Foot
39. Armoured Muslim Lancers (Superior)
40. Dismounted (Superior) Armoured Muslim
    Lancers
41. Armoured Muslim Lancers (Average)
42. Dismounted (Average) Armoured Muslim Lancers
43. Muslim Lancers
44. Ghilman
45. Dismounted Ghilman
46. Muslim Cavalry (Light Spear)
47. Muslim War Elephants
48. Muslim Light Horse (Javelins)
49. Muslim Light Foot Archers
50. Muslim Light Javelinmen
51. Naffatun
52. Veteran Dailami Foot
53. Dailami Foot
54. Superior Indian Lancers
55. Indian Lancers
sentence_splits_diff: 70.00;
       fwedu_hw:No of Right Mag. the ascension
            Declination of the Stars deg ' " d '
            " Stars 1 149 48 55 6 7 45 S 6 A new
            Star, the Comet compared May 18 and
            19 at night 22 151 25 27 6 52 25 6 A
            Star of the Sextant, Comet compared
            May 14, 15, 16, and 17 2 153 25 25 5
            50 36 7 A new Star, Comet compared
            May 17, 18, 19, and 20 3 153 33 8 5
            12 52 8 A new Star, Comet compared
            May 20, 21, 22, 23, and 24 27 153 35
            49 3 10 3 6 A Star of the Sextant,
            Comet compared June 3 4 153 42 34 4
            28 35 10 A new Star, Comet compared
            May 24, 25, 26, 27, and 28 5 153 42
            34 5 21 5 10 A new Star, Comet
            compared May 20 6 153 46 21 7 58 55 8
             A new Star, Comet compared May 14 7
            153 47 22 7 12 32 10 A new Star,
            Comet compared May 14 8 153 57 40 8
            19 29 10 A new Star, Comet compared
            May 13 9 154 7 39 4 13 20 10 A new
            Star, Comet compared May 30 10 154 47
             30 12 21 11 7 A new Star, Comet
            compared May 7 11 154 52 21 11 52 46
            9 A new Star, Comet compared May 8 12
             154 57 2 13 39 6 7 A new Star, Comet
             compared May 6 1 156 8 56 15 5 53 6
            A Star of the Hydra Phi2, Comet
            compared May 5 13 156 8 15 11 1 27 7
            A new Star, Comet compared May 9 2
            156 43 56 15 37 38 5 Phi3 of the
            Hydra, Comet compared May 5 14 156 58
             8 19 36 31 9 A new Star, Comet
            compared May 13 15 157 40 41 9 21 39
            8 A new Star, Comet compared May 3 16
             159 25 32 8 34 41 6 A new Star,
            Comet compared May 12 17 159 26 5 25
            31 55 7 A new Star, Comet compared
            May 1 17.1 160 0 28 25 36 24 9 A new
            Star 18 322 3 15 20 52 19 8 A new
            Star, Comet compared April 17 in the
            Morning 19 322 25 3 20 41 56 7 A new
            Star, Comet compared April 17 in the
            Morning 49 323 25 38 17 12 5 3 Delta
            of Capricorn, Comet estimated April
            14 and 15 in the Morning 20 325 10 5
            11 26 3 7 A new Star, Comet compared
            April 7 and 8 in the Morning 21 326
            12 50 9 42 0 7 A new Star, Comet
            compared April 6 in the Morning 30
            327 41 29 7 40 52 6 A Star of
            Aquarius, Comet compared April 1 in
```

```
           the Morning 22 329 11 24 8 41 31 7 A
           new Star, Comet compared April 2 in
           the Morning 5 344 4 32 0 48 0 N 6 A
           of Pisces, Comet compared and
           estimated February 14 23 348 33 20 1
           7 28 8 A new Star, Comet compared
           January 31 and February 1 8 348 38 36
            0 3 40 S 5 Kappa' of Pisces, Comet
           compared February 3 and 4 24 348 49
           10 0 32 29 N 8 A new Star, Comet
           compared February 1 25 349 40 54 1 1
           14 8 A new Star, Comet compared
           January 27 and 28 26 350 44 16 0 50
           56 10 A new Star, Comet compared
           January 25 27 350 52 16 1 5 8 10 A
           new Star, Comet compared January 23
           16 351 0 1 0 44 48 6 A Star of Pisces
           , Comet estimated January 25, 27, and
            28 28 352 13 5 1 6 40 8 A new Star,
           Comet compared January 21 and 22 18
           352 26 6 0 25 53 5 Lambda of Pisces,
           Comet estimated January 21
       fwedu_simp:1. 149 48 55 6 7 45 S A new
           Star was seen on May 18 and 19 at
           night.
2. 153 25 27 6 52 25 6 A Star of the Sextant was
     seen on May 14, 15, 16, and 17.
3. 153 33 8 5 12 52 8 A new Star was seen on May
     17, 18, 19, and 20.
4. 153 35 49 3 10 3 6 A Star of the Sextant was
    seen on June 3 and 4.
5. 153 42 34 5 21 5 10 A new Star was seen on
    May 24, 25, 26, 27, and 28.
6. 153 46 21 7 58 55 8 A new Star was seen on
    May 14.
7. 153 47 22 7 12 32 10 A new Star was seen on
    May 14.
8. 153 57 40 8 19 29 10 A new Star was seen on
    May 13.
9. 154 7 39 4 13 20 10 A new Star was seen on
    May 30.
10. 154 47 30 12 21 11 7 A new Star was seen on
    May 7.
11. 154 52 21 11 52 46 9 A new Star was seen on
    May 8.
12. 154 57 2 13 39 6 7 A new Star was seen on
    May 6.
13. 156 8 56 15 5 53 6 A Star of the Hydra Phi2
    was seen on May 5.
14. 156 8 15 11 1 27 7 A new Star was seen on
    May 9.
15. 156 43 56 15 37 38 5 Phi3 of the Hydra was
    seen on May 5.
16. 156 58 8 19 36 31 9 A new Star was seen on
    May 13.
17. 157 40 41 9 21 39 8 A new Star was seen on
    May 3.
18. 159 25 32 8 34 41 6 A new Star was seen on
    May 12.
19. 159 26 5 25 31 55 7 A new Star was seen on
    May 1.
20. 160 0 28 25 36 24 9 A new Star was seen.
21. 322 3 15 20 52 19 8 A new Star was seen on
    April 17 in the Morning.
22. 322 25 38 17 12 5 3 Delta of Capricorn was
    seen on April 14 and 15 in the Morning.
23. 325 10 5 11 26 3 7 A new Star was seen on
    April 7 and 8 in the Morning.
24. 326 12 50 9 42 0 7 A new Star was seen on
    April 6 in the Morning.
25. 327 41 29 7 40 52 6 A Star of Aquarius was
    seen on April 1 in the Morning.
26. 329 11 24 8 41 31 7 A new Star was seen on
    April 2 in the Morning.
27. 344 4 32 0 48 0 N A of Pisces was seen and
    estimated on February 14.
28. 348 33 20 1 7 28 8 A new Star was seen on
    January 31 and February 1.
29. 348 38 36 0 3 40 S Kappa' of Pisces was seen
     on February 3 and 4.
30. 348 49 10 0 32 29 N A new Star was seen on
    February 1.
31. 349 40 54 1 1 14 8 A new Star was seen on
    January 27 and 28.
32. 350 44 16 0 50 56 10 A new Star was seen on
    January 25.
33. 350 52 16 1 5 8 10 A new Star was seen on
    January 23.
34. 351 0 1 0 44 48 6 A Star of Pisces was seen
    and estimated on January 25, 27, and 28.
35. 352 13 5 1 6 40 8 A new Star was seen on
    January 21.
36. 352 26 6 0 25 53 5 Lambda of Pisces was seen
     and estimated on January 21.
```

Contrastive Decoding for Synthetic Data Generation in Low-Resource Language Modeling

Jannek Ulm[1] **Kevin Du**[1] **Vésteinn Snæbjarnarson**[1,2]
[1]ETH Zürich [2]University of Copenhagen
jannek.ulm@gmail.com kevin.du@inf.ethz.ch vest.snae@gmail.com

Abstract

Large language models (LLMs) are trained on huge amounts of textual data, and concerns have been raised that the limits of such data may soon be reached. A potential solution is to train on synthetic data sampled from LLMs. In this work, we build on this idea and investigate the benefits of *contrastive decoding* for generating synthetic corpora. In a controlled setting, we experiment with sampling corpora using the relative difference between a GOOD and BAD model trained on the same original corpus of 100 million words. By amplifying the signal from a model that has better performance, we create a synthetic corpus and mix it with the original training data. Our findings show that training on a mixture of synthesized and real data improves performance on the language modeling objective and a range of downstream tasks. In particular, we see that training with a mix of synthetic data from contrastive decoding benefits tasks that require more *reasoning skills*, while synthetic data from traditional sampling helps more on tasks dependent on surface-level *linguistic* capabilities.

https://github.com/janulm/CD-for-Synthetic-Data-Generation

1 Introduction

Large language models (LLMs) require enormous amounts of text to achieve strong performance (Kaplan et al., 2020; Hoffmann et al., 2022). For the largest models, it has even been claimed that current training regimes already consume the vast majority of publicly available text on the internet (Villalobos et al., 2024; Dubey et al., 2024). The BabyLM Challenge (Charpentier et al., 2025) emphasizes this point by asking what can be learned under a strict budget of 100M words, prioritizing data efficiency over raw scale, mimicking the far more efficient language learning capabilities of humans. Furthermore, not all training data is equally beneficial (Eldan and Li, 2023; Gunasekar et al., 2023). The question thus arises: How can we get more high-quality data in a constrained setting? One proposed solution is to generate synthetic data using existing pre-trained models, thereby expanding the available corpus without collecting more human-written text (Wang et al., 2023; Eldan and Li, 2023; Gunasekar et al., 2023; Abdin et al., 2024).

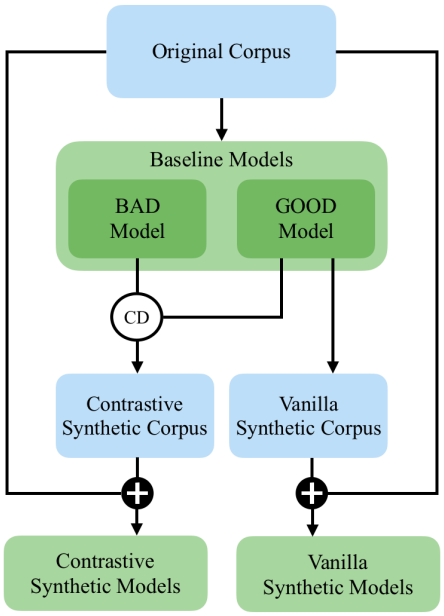


Figure 1: Our synthetic data generation and training pipeline: Start by training baseline LMs on a "real" corpus (*TinyBabyLM*: human-written text + *TinyStories*). The GOOD model is the best checkpoint; the BAD model is a weaker variant, e.g., an earlier checkpoint. We generate synthetic corpora via (i) *contrastive decoding (CD)*, and (ii) non-contrastive ancestral (*vanilla*) sampling. We then train new models on a mixture of the original and synthetic corpora. We find that contrastive models improve the most over the BASELINE in evaluations on reasoning-oriented benchmarks, such as entity tracking.

Generating synthetic data is non-trivial, however. The quality of synthetic text may be hindered by

Proceedings of the First BabyLM Workshop, pages 29–41
November 8, 2025

noise, factual errors, or stylistic artifacts (Lin et al., 2022; Huang et al., 2025). Models may also replicate or even amplify biases from their training data (Gallegos et al., 2024; Bender et al., 2021), and generated text may diverge from the target distribution, leading to potential degradation in downstream performance or model collapse (Dohmatob et al., 2025; Gerstgrasser et al., 2024; Shumailov et al., 2024). Moreover, producing high-quality synthetic data is particularly difficult because language models often hallucinate facts or repeat memorized content from their original training corpus (Bender et al., 2021; Lin et al., 2022).

This work explores the use of contrastive decoding (CD) (Li et al., 2023) to generate synthetic data in a controlled setting. CD is a decoding strategy that takes advantage of the differences between a GOOD model and a BAD model to produce more coherent and informative text. In prior work, CD has been largely restricted to improving the quality of responses generated for inference-time tasks (Li et al., 2023; O'Brien and Lewis, 2023; Chang et al., 2024). In contrast, we use CD to synthesize corpora to train new models from scratch. Our goal is to know whether these inference-time benefits of CD translate into gains when generating synthetic data for training language models.

The high-level experimental approach is illustrated in Figure 1 and goes as follows.

1. Start with an original corpus (100M tokens, BabyLM setting (Charpentier et al., 2025)).
2. Train BASELINE models (100M-parameter models based on the Llama 2 architecture (Touvron et al., 2023)) on the original corpus.
3. Generate synthetic corpora (100M tokens each) using CD and standard sampling.
4. Train models on the original and synthetic corpora.
5. Evaluate models on downstream tasks and compare to BASELINE.

We find that synthetic data improves performance on the language-modeling objective and downstream tasks. Moreover, tasks that emphasize reasoning benefit most from CD-generated data, whereas tasks emphasizing linguistic competence gain more from standard (non-contrastive) sampling.

2 Synthetic Data Generation for Pre-training Language Models

Recent work shows that *curated, high-quality* synthetic corpora can substantially boost data efficiency for small or low-resource LMs (Eldan and Li, 2023). Carefully constructed "textbook"-style corpora improve generalization (Gunasekar et al., 2023), and iterative pipelines that generate, critique, and revise synthetic content have been shown to boost reasoning-oriented capabilities (Abdin et al., 2024). Domain-targeted corpora can be especially effective: *TinyStories* demonstrates that fully synthetic, child-directed narratives enable 1–10M-parameter models to produce multi-paragraph coherent and grammatical text (Eldan and Li, 2023). For instruction following, Self-Instruct bootstraps instruction-response pairs from a seed set, leading to gains without additional human annotation (Wang et al., 2023). These results collectively suggest that synthetic data can significantly increase downstream performance.

However, naive reuse of model-generated text across generations can severly harm performance, resulting in "model collapse" (Shumailov et al., 2024; Gerstgrasser et al., 2024; Dohmatob et al., 2025). Empirically, careful filtering, diversification, and sustained mixing with real data mitigate such risks while preserving gains (Gerstgrasser et al., 2024). In this work, we explore an orthogonal axis: *decoding-control* for synthetic corpora. Specifically, we study whether CD can produce higher-signal synthetic corpora for pre-training under a tight data budget, compared to non contrastive approaches.

3 Contrastive Decoding

Language-models. Following Cotterell et al. (2024), let Σ be a set of tokens we call the vocabulary, the Kleene closure Σ^* be the set of all strings built from Σ, if p is a probability distribution over Σ^* we say it is a **language model**. Then, $p(x_i \mid \boldsymbol{x}_{<i})$ represents the model's *next-token* probability, i.e., the probability that the next token is x_i given the preceding context $\boldsymbol{x}_{<i} \stackrel{\text{def}}{=} x_0 x_1 \ldots x_{i-1}$.

Contrastive Decoding We now describe the CD approach in detail. Let p_{G} be a GOOD (better performing) language model, and p_{B} be a BAD (worse performing) language model. Following Li et al. (2023), we define $\mathcal{V}_{\text{head}}$ as the set of likely

tokens under p_{G}:

$$\mathcal{V}_{\text{head}}\left(\boldsymbol{x}_{<i}\right) \stackrel{\text{def}}{=} \{x_i \in \Sigma : \quad (1)$$
$$p_{\mathrm{G}}\left(x_i \mid \boldsymbol{x}_{<i}\right) \geq \alpha \max_{w \in \Sigma} p_{\mathrm{G}}\left(w \mid \boldsymbol{x}_{<i}\right)\}.$$

Where α is a scalar hyper-parameter. The contrastive score CD for $x_i \in \mathcal{V}_{\text{head}}\left(\boldsymbol{x}_{<i}\right)$ is then defined as follows:

$$\mathsf{CD}\left(x_i \mid \boldsymbol{x}_{<i}\right) \stackrel{\text{def}}{=} \quad (2)$$
$$\log p_{\mathrm{G}}\left(x_i \mid \boldsymbol{x}_{<i}\right) - \lambda \log p_{\mathrm{B}}\left(x_i \mid \boldsymbol{x}_{<i}\right),$$

where contrast strength is controlled by a scalar λ. Further, if $x_i \notin \mathcal{V}_{\text{head}}\left(\boldsymbol{x}_{<i}\right)$ then $\mathsf{CD}\left(x_i \mid \boldsymbol{x}_{<i}\right) \stackrel{\text{def}}{=} -\inf$. Typically, the contrastive scores $\mathsf{CD}\left(\cdot \mid \boldsymbol{x}_{<i}\right)$ are treated as logits giving rise to a new probability distribution over Σ from which we can decode the next token.

Background and variants. CD biases generation toward tokens preferred by a stronger GOOD model while down-weighting those preferred by a weaker BAD model, under the plausibility mask $\mathcal{V}_{\text{head}}$ (Li et al., 2023). Empirically, CD reduces repetition and topic drift in open-ended generation and, without additional training, improves reasoning-focused decoding compared to greedy or nucleus (top-p) sampling (Li et al., 2023; O'Brien and Lewis, 2023).

Several works adapt CD to lower its compute and memory cost or to strengthen specific capabilities. Phan et al. (2024) replace an explicit bad model with a distilled proxy (e.g., via dropout or quantization), retaining most of CD's gains while reducing memory. In retrieval or context-heavy settings, Zhao et al. (2024) integrate CD with adversarial negatives so that decoding remains grounded in relevant passages. These methods focus on evaluating the CD-like inference performance, rather than on generating pre-training corpora.

Relation to synthetic-data generation. A related approach is STEER, which performs contrastive expert guidance by subtracting a base model from a fine-tuned domain expert and combining it with negative prompting to generate synthetic corpora for downstream fine-tuning (O'Neill et al., 2023). In contrast, we use CD with a general GOOD/BAD pair trained on the same base corpus and treat CD as a data generator for pre-training: we synthesize full corpora and then train new models from scratch on mixtures of real and synthetic text. This lets us test whether CD's inference-time benefits translate into better pre-training signals, and how they compare to vanilla sampling under a fixed data budget.

4 Training on Synthetic Data

Given the success of CD in generating higher scoring text for evaluations, we ask whether it can also be employed to generate higher-quality text for pre-training. This section describes our procedure for generating synthetic corpora using CD and training models on them.

4.1 Synthetic Corpus Generation

General Procedure. To ensure independence from the training data, following (Wang et al., 2023) we generate synthetic corpora from *prefix seeds* that are held out from all training and evaluation data. The prefix seeds are evenly sampled across the four data sources to preserve balance, we describe this in more detail in Section 5.1. For each prefix seed, we fix the first 20 tokens as a context prefix, and then we sample continuations from the target model. To ensure sufficient diversity and corpus size, we produce eight completions of up to 400 tokens per seed. To sample each next token, we use the decoding strategies described below. Using ∼30.4K generation seeds we produce approximately 100M tokens for each decoding strategy.

Decoding Strategies. We mainly compare two decoding settings that differ only in how candidate tokens are scored before sampling. Let $\mathcal{V}_{\text{head}}(\boldsymbol{x}_{<i})$ be the set of α-likely tokens of the GOOD distribution p_{G} as defined in Eq. (1); If $\mathcal{V}_{\text{head}}(\boldsymbol{x}_{<i})$ is applied, tokens outside $\mathcal{V}_{\text{head}}$ are assigned score $-\infty$ (Li et al., 2023). Let the contrastive score $\mathsf{CD}(x_i \mid \boldsymbol{x}_{<i})$ be as in Eq. (2). For CD we treat $\mathsf{CD}(\cdot \mid \boldsymbol{x}_{<i})$ as a logit over $\mathcal{V}_{\text{head}}(\boldsymbol{x}_{<i})$, i.e., we sample with probabilities proportional to $\exp\big(\mathsf{CD}(x_i \mid \boldsymbol{x}_{<i})\big)$.

1. **NO-CONTRAST:** Ancestral sampling from $p_{\mathrm{G}}\left(\cdot \mid \boldsymbol{x}_{<i}\right)$.
2. **CONTRASTIVE DECODING (CD):** Ancestral sampling within $\mathcal{V}_{\text{head}}(\boldsymbol{x}_{<i})$ using logits $\mathsf{CD}(x_i \mid \boldsymbol{x}_{<i})$ (Eq. (2)), which promote tokens preferred by p_{G} over p_{B}.

We also study the effect of truncating the sampling support to further suppress low-probability continuations as follows:

3. **NO-CONTRAST + $\mathcal{V}_{\text{head}}$:** Ancestral sampling from $p_{\mathrm{G}}(\cdot \mid \boldsymbol{x}_{<i})$ restricted to $\mathcal{V}_{\text{head}}(\boldsymbol{x}_{<i})$.

4. **NO-CONTRAST + top-p:** Ancestral sampling from $p_{\mathrm{G}}(\cdot \mid \boldsymbol{x}_{<i})$ restricted to top-p selection (Holtzman et al., 2020).

5. **NO-CONTRAST + top-k:** Ancestral sampling from $p_{\mathrm{G}}(\cdot \mid \boldsymbol{x}_{<i})$ restricted to top-k selection (Fan et al., 2018).

6. CD **with top-p:** Ancestral sampling restricted to the top-p after already restricting to the $\mathcal{V}_{\text{head}}(\boldsymbol{x}_{<i})$ using logits $\mathsf{CD}(x_i \mid \boldsymbol{x}_{<i})$ (Eq. (2)).

7. CD **with top-k:** Ancestral sampling restricted to the top-k after already restricting to the $\mathcal{V}_{\text{head}}(\boldsymbol{x}_{<i})$ using logits $\mathsf{CD}(x_i \mid \boldsymbol{x}_{<i})$ (Eq. (2)).

We sweep $k \in \{50, 100, 200\}$ and $p \in \{0.90, 0.95, 0.97\}$ and report effects on performance in Section 6.4 and Table 4.

4.2 The BAD and GOOD Models

We consider three approaches to instantiate a BAD model p_{B} (details in Appendix A):

i) **Smaller models** that are $10\times$, $20\times$, $50\times$, and $100\times$ smaller than the GOOD model, and, following (Li et al., 2023), selecting the checkpoint with the best evaluation perplexity.

ii) **Earlier checkpoints**, e.g., if a GOOD checkpoint is taken at step 2500, we test BAD checkpoints at steps 2000, 1500, 1000 and 500.

iii) **Attention dropout**, where the BAD model is the GOOD model, but run with attention dropout rates $\{0.1, 0.3, 0.5, 0.7\}$ at inference time (Phan et al., 2024).

Note on scale. Prior evaluations of CD use billion-parameter GOOD models paired with much smaller BAD models (e.g., OPT-13B vs. OPT-125M; GPT-2-XL vs. GPT-2-small), and report that performance improves as the GOOD–BAD *scale gap* increases (Li et al., 2023, §7.1; Fig. 2). CD is not limited to GOOD models that are several billion parameters or larger, e.g., as Li et al. (2023) also show gains with GPT-2-XL ($\sim$1.5B). However, the observed size-gap effect suggests that a strong contrast may be harder to elicit at our $\sim$100M-parameter scale. Consistent with this, O'Brien and Lewis (2023) find that smaller BAD models help more than larger ones and that gains tend to be stronger for larger GOOD models on reasoning tasks. We therefore investigate multiple BAD model instantiations to identify how we can elicit a sufficient contrastive signal at this scale (Li et al., 2023; O'Brien and Lewis, 2023; Phan et al., 2024).

Hyperparameters. Following Li et al. (2023), we use $\alpha = 0.1$ for $\mathcal{V}_{\text{head}}$ and the contrast strength is set to $\lambda = 1$.

The GOOD models. We describe how the better models, p_{G}, are selected in Section 5.2.

4.3 Training with Mixed Corpora

All models are trained from scratch to isolate the effect of the synthetic corpora. For each decoding method, we mix its 100M-token synthetic corpus with the same 100M-token TinyBabyLM corpus (see Section 5.1) used for the baselines, while keeping initialization seeds, training length, and optimization hyperparameters identical to the baseline runs. Batches contain 256 sequences of 1024 tokens with a fixed 70/30 mixture at the sequence level (70% real, 30% synthetic) and are repeatedly regrouped and re-tokenized to act as a data regularizer (see Section 5.2 and Appendix A).

We ablate the original/synthetic mixture and report its effect on performance in Section 6.5. Since initial testing indicated that the 70/30 mixture achieved the strongest average performance across tasks, we report results under this fixed ratio in the main experiments.

5 Experimental Details

5.1 *TinyBabyLM* Corpus

We start from the BabyLM 100M corpus and construct a modified variant by replacing the CHILDES, BNC and SWITCHBOARD portions with the synthetic *TinyStories* (Eldan and Li, 2023). We add a portion of *TinyStories* because Eldan and Li (2023) show that their corpus, a constrained, child-directed synthetic corpus enables very small models (1–10M parameters) to learn fluent, grammatical multi-paragraph stories, making it a high-signal, data-efficient addition for low-resource pretraining. Concretely, we substitute $\sim$39.7M words of TinyStories for the removed words, yielding the following composition: Gutenberg (27.4M), SimpleWiki (14.9M), OpenSubtitles (17.7M) and TinyStories (39.7M) (Eldan and Li, 2023; Gerlach and Font-Clos, 2020; Lison and Tiedemann, 2016). We refer

to this modified corpus as *TinyBabyLM*. The total amount of human-written+TinyStories text is held at ≈100M words; note that "words" here denote whitespace-delimited tokens, so totals differ from BPE token counts used during training (see Section A). We partition TinyBabyLM into three disjoint splits: *train* (90.5M words), *eval* (8.9M words), and *seeds* (600K words). The generation seeds (a selection of ∼30K paragraph start prefixes) for synthetic generation are sampled exclusively from the *seeds* split and are strictly disjoint from all *train* and *eval* text. To maintain balance across domains, the splits, *seed*, *train* and *eval*, are distributed evenly across the four data sources.

5.2 Model Architecture & Training Setup

We use a decoder-only Transformer LLaMA-2 architecture with ∼100M parameters (Touvron et al., 2023): 12 layers, hidden size 768, 12 attention heads, MLP intermediate size 3072, and a maximum context length of 1024 tokens. All models are trained from scratch with the same initialization scheme.

Tokenization is performed with a SentencePiece BPE tokenizer (vocabulary size 32k) trained on the TinyBabyLM corpus; the same tokenizer is used for all experiments to ensure comparability (see Appendix A for details).

Training uses a global batch of 256 sequences × 1024 tokens, AdamW with weight decay 0.1, and a cosine learning-rate schedule: peak 1×10^{-3}, 150 warm-up steps, and decay to zero by step 8000. The training duration is fixed to 8000 steps for every run and checkpoints are saved every 500 steps. Each experimental condition is repeated with $n = 10$ distinct random seeds.

Data pipeline (applies to all runs). Real and synthetic corpora are stored as rows of text and, at the start of training, are independently shuffled, tokenized, and split into fixed-length sequences. Sampling proceeds until a corpus is exhausted, at which point that corpus is reshuffled, and re-segmented before resuming. This periodic resegmentation acts as a regularizer and is applied identically to baseline and mixed-data runs.

Good checkpoint selection. From each of the $n = 10$ BASELINE seeds, we first select the saved checkpoint with the lowest perplexity, forming the candidate set $\mathcal{X}$. We then evaluate only $\mathcal{X}$ on the full suite of tasks, convert scores to percentiles within the task, average percentile across tasks, and choose as the GOOD model the checkpoint with the highest average percentile.

5.3 Evaluation & Statistical Analysis

Benchmarks. We evaluate on the zero-shot BabyLM evaluation suite[1] and report Perplexity on the TinyBabyLM *eval*-split (see 5.1). The tasks considered are:

- **BLiMP**: Benchmark of Linguistic Minimal Pairs testing core English grammar linguistic competence (Warstadt et al., 2020).
- **BLiMP Supplement**: BLiMP-style suite, extending to dialogue and question answering, focused on reasoning, syntax and semantics (Hu et al., 2024; Warstadt et al., 2023).
- **EWoK**: Checks for social/physical/world knowledge and semantic understanding (Ivanova et al., 2024).
- **Entity Tracking**: Requires maintaining and updating entity states across text to test memory and state reasoning (Kim and Schuster, 2023).
- **WUG**: Evaluates morphology, evaluating on adjective nominalization to estimate linguistic generalization (Hofmann et al., 2025).
- **Reading**: Compares model surprisal to human word-by-word reading times to assess processing alignment (De Varda et al., 2023).
- **Eye-Tracking**: Tests whether model predictability tracks human eye-movement measures during reading (De Varda et al., 2023).

The metric used for the **Reading** and **Eye-tracking** tasks is the partial change (%) in the coefficient of determination, that is, the additional proportion of variance explained. For the other tasks, accuracy is used.

Per-task mean-max over checkpoints. For each training method[2] m, benchmark task t, and initialization seed s, we save checkpoints every 500 steps and select the best checkpoint independently per (m, t, s). Let $\mathcal{C}_{m,t,s}$ denote the set of saved checkpoints over the steps, and $S_{m,t,s}(c)$ the task score at checkpoint c. For higher-is-better tasks, we set

$$c^*_{m,t,s} \stackrel{\text{def}}{=} \arg\max_{c \in \mathcal{C}_{m,t,s}} S_{m,t,s}(c),$$

[1] https://github.com/babylm/evaluation-pipeline-2025

[2] Either BASELINE, or a pair of decoding strategy from 4.1 and bad model setting from 4.2

Name	Perplexity↓	BLiMP↑	BLiMP Supp.↑	Entity Tracking↑	EWoK↑	WUG↑	Reading↑	Eye Tracking↑
GOOD	24.62	71.22	63.50	27.01	53.64	57.50	1.44	3.51
BASELINE	24.46±0.10	71.03±0.27	64.10±0.60	27.82±1.18	53.18±0.28	66.90±2.47	1.76±0.22	3.85±0.31

Table 1: Reference performance of the BASELINE (mean ± s.e., n=10 independent runs; per-task mean–max checkpointing per Section 5.3) versus the single fixed GOOD checkpoint. Because it is a single checkpoint chosen once across seeds rather than per task, it can sit below the BASELINE mean on some tasks.

while for perplexity we take $\arg\min$. The selected checkpoint $c^*_{m,t,s}$ is then evaluated. This procedure estimates the best attainable performance per task under the fixed training budget and avoids coupling to a single global checkpoint.

Paired bootstrap for statistical significance. Evaluation of checkpoint $c^*_{m,t,s}$ for task t yields per-example outcomes $y_{m,t,s,i}$ for examples $i = 1, \ldots, N_t$. We use paired bootstrap with $B = 1000$ resamples to calculate confidence intervals. For each (t, s) and bootstrap draw b, sample the index-set $I^{(b)}$ of size N_t with replacement from $\{1, \ldots, N_t\}$ and apply the *same* $I^{(b)}$ to all methods (pairing). We average out uncertainty over the seeds:

$$\bar{y}^{(b)}_{m,t,s} = \frac{1}{N_t} \sum_{i \in I^{(b)}} y_{m,t,s,i} \tag{3}$$

$$\mu^{(b)}_{m,t} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \bar{y}^{(b)}_{m,t,s}. \tag{4}$$

As in (3), $\bar{y}^{(b)}_{m,t,s}$ is the mean for tasks with per-example scalar scores (e.g., BLiMP, EWoK). For metrics with task-specific aggregations (e.g., Perplexity or Reading), we substitute the appropriate aggregation function and proceed identically. For a comparison of two methods m_1 and m_2, we form the bootstrap difference distribution

$$\Delta^{(b)}_t = \mu^{(b)}_{m_1,t} - \mu^{(b)}_{m_2,t} \tag{5}$$

We compute 95% confidence intervals via the percentile method, $\mathrm{CI}_{95} = [\mathrm{pct}_{2.5}, \mathrm{pct}_{97.5}]$ of $\{\Delta^{(b)}_t\}_{b=1}^B$. A difference is deemed significant if $0 \notin \mathrm{CI}_{95}$. We compute one-sided p-values in the direction of the observed effect using the estimator on the bootstrap differences $\{\Delta^{(b)}_t\}_{b=1}^B$: for higher-is-better tasks with $\hat{\Delta}_t > 0$,

$$p = \frac{1 + \sum_{b=1}^B \mathbb{I}\{\Delta^{(b)}_t \le 0\}}{B + 1} \tag{6}$$

and if $\bar{\Delta}_t < 0$ use $\ge$ instead. For lower-is-better metrics we swap the inequality accordingly.

Aggregate reporting. For tables and figures, we bold the best method per benchmark and mark significant improvements/degradations relative to the BASELINE. We report, for each method m and task t, the bootstrap mean $\bar{\mu}_{m,t}$ and standard-error.

$$\bar{\mu}_{m,t} = \frac{1}{B} \sum_{b=1}^B \mu^{(b)}_{m,t}, \quad \widehat{\mathrm{SE}}_{m,t} = \frac{\sigma_{m,t}}{\sqrt{B}} \tag{7}$$

This analysis serves to estimate the maximum achievable performance for each method, on each task, given the training setup. Our aggregating metric $\mu_{\Delta\mathrm{REL}}$ is the mean relative performance, across all tasks except Perplexity, vs. the BASELINE—i.e., it is the average proportional change given in percentages.

6 Results

6.1 BASELINE Performance

Table 1 summarizes the performance of our reference points, the GOOD and BASELINE results. Recall that the BASELINE row reports the mean ± s.e. over n=10 independent runs under our per-task bootstrapped mean–max evaluation (Section 5.3). In contrast, the GOOD model is a *single* checkpoint selected once, across seeds, using the selection procedure described in Section 5.2. As such GOOD is broadly representative of a strong model but sits slightly below the BASELINE mean on some tasks (e.g., Perplexity, BLiMP Supplement) because it cannot adapt per task. We use this fixed checkpoint as the GOOD model in all subsequent synthetic corpora and comparisons.

6.2 Contrastive vs. non-contrastive generation

Setup. Recall from Section 4.1 that we compare the three generation settings for synthesizing the 100M-token corpora: (i) NO-CONTRAST, (ii) NO-CONTRAST–$\mathcal{V}_{\mathrm{HEAD}}$, and (iii) CD . Among all contrastive instantiations, using the early checkpoint at 500 steps (**CD–Early–500**) emerged as the strongest (see Section 6.3), and we use it as our CD representative in this section. Results are summarized in Table 2.

Name	$\mu_{\Delta\text{REL}}$ ↑	Perplexity↓	BLiMP↑	BLiMP Supp.↑	Entity Tracking↑	EWoK↑	WUG↑	Reading↑	Eye Tracking↑
Baseline	-	24.46±0.10	71.03±0.27	64.10±0.60	27.82±1.18	53.18±0.28	66.90±2.47	1.76±0.22	3.85±0.31
No-Contrast	2.96%	**23.56±0.11***	**72.09±0.17***	64.83±0.73	28.14±1.75	53.17±0.30	64.67±1.66*	**1.91±0.25**	4.31±0.33*
No-Contrast-V-Head	0.66%	24.33±0.10*	71.67±0.24*	64.86±0.74	25.47±1.40*	53.03±0.31	66.67±1.58	1.76±0.23	4.32±0.33*
CD-Early-500	**4.90%**	23.73±0.10*	71.72±0.19*	**65.10±0.60***	**30.38±0.65***	**53.80±0.29***	**70.55±2.32***	1.79±0.22	**4.42±0.32***

Table 2: Task-by-task results for synthetic-data regimes. Entries are mean ± s.e.; * denotes a significant difference vs. BASELINE. CD-Early-500 attains the best overall $\mu_{\Delta\text{REL}}$ (+4.90%) and leads on BLiMP Supplement, Entity Tracking, EWoK, WUG, and Eye Tracking, while NO-CONTRAST yields the lowest Perplexity, the best BLiMP and Reading. Find relative change vs. BASELINE at Table 6

Aggregate performance. All synthetic regimes beat BASELINE. CD delivers the strongest overall gains ($\mu_{\Delta\text{REL}}$ +4.90%), with the non-contrastive variants lacking, see Table 2.

Language modeling (Perplexity). Perplexity drops for every method. NO-CONTRAST attains the lowest value (23.56), with CD close behind, so non-contrastive sampling edges out CD slightly on the LM objective, while CD still clearly improves over BASELINE; see Table 2.

Metric	CD vs NO-CONTRAST	Significance
$\mu_{\Delta\text{REL}_{\text{CD}}} - \mu_{\Delta\text{REL}_{\text{NO-CONTRAST}}}$	+1.94pp	
Perplexity↓	-0.7%	***
BLiMP↑	-0.5%	***
BLiMP Supp.↑	+0.4%	
Entity Tracking↑	+7.3%	***
EWoK↑	+1.2%	*
WUG↑	+8.2%	***
Reading↑	-6.2%	
Eye Tracking↑	+2.5%	

Table 3: Statistical significance and relative change of CD–EARLY–500 vs. NO-CONTRAST by metric. Entries are percentage changes; for Perplexity (↓), more negative is better, while for all others (↑), more positive is better. The "Significance" column reports paired-bootstrap one-sided p-values per Section 5.3: * $p<0.05$, ** $p<0.01$, *** $p<0.001$ (blank = not significant). $\mu_{\Delta\text{REL}}$ is shown as an absolute difference in percentage points (pp).

Task-level pattern and head-to-head. CD performs best on five tasks and shows significant gains on five, notably on reasoning-/tracking-oriented evaluations like BLiMP Supplement, Entity Tracking, and EWoK (see Table 2. In contrast, NO-CONTRAST is best on three tasks with significant effects on four, and it leads on core linguistic competence with Perplexity and BLiMP. In direct statistical comparisons (CD vs. NO-CONTRAST), as displayed in Table 3, NO-CONTRAST has a small but significant edge on Perplexity and BLiMP, whereas CD achieves significant, and generally larger, gains on Entity Tracking, EWoK, and WUG. The remaining tasks show no reliable difference.

Is it the $\mathcal{V}_{\text{head}}$ mask or the contrastive logits? NO-CONTRAST–$\mathcal{V}_{\text{HEAD}}$ serves as a control that isolates the effect of restricting to the α-head without any contrastive subtraction. If head-masking alone explained CD's gains, NO-CONTRAST–$\mathcal{V}_{\text{HEAD}}$ would mirror CD. It does not: while NO-CONTRAST–$\mathcal{V}_{\text{HEAD}}$ modestly helps Perplexity (–0.51%) and Eye Tracking (+12.12%), it significantly hurts Entity Tracking (–8.45%) and yields small/neutral changes elsewhere (Table 2). This suggest that the improvements are driven by the contrastive logits and not the $\mathcal{V}_{\text{head}}$ constraint.

Takeaway. Mixing synthetic data consistently helps. Among generation strategies, CD delivers the strongest overall improvements and a clear advantage on reasoning-oriented benchmarks, while NO-CONTRAST remains best for the LM objective and BLiMP. The NO-CONTRAST–$\mathcal{V}_{\text{HEAD}}$ control suggests that contrastive scoring, not head-masking, is the key to CD's benefits.

Name	$\mu_{\Delta\text{REL}}$ ↑	Perplexity↓
BASELINE	-	24.46±0.10
NO-CONTRAST	2.96%	**23.56±0.11*** (3.68%)
NO-CONTRAST-Top-k-200	3.65%	23.65±0.10* (3.29%)
CD-Small-20	3.55%	23.73±0.14* (2.96%)
CD-Drop-0.7	3.29%	24.06±0.13* (1.65%)
CD-Early-500	4.90%	23.73±0.10* (2.98%)
CD-Early-500-Top-k-200	**5.69%**	23.77±0.10* (2.80%)

Table 4: Comparison of CD variants (early checkpoint, smaller model, dropout) against non-contrastive baselines, including the best truncation configurations. The best truncation for both regimes is Top-k=200; CD-Early-500-Top-k-200 achieves the highest overall task improvement at unchanged perplexity.

6.3 Searching for Effective CD Settings

We instantiate the amateur for contrastive decoding using three settings (Section 4.2): (i) earlier checkpoints, (ii) smaller models, and (iii) inference-time attention dropout. We report the best setting from each setting in Table 4 and the full results in Table 6

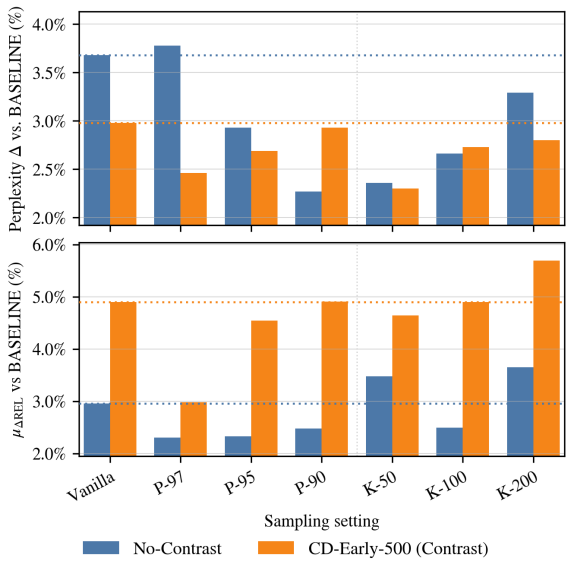


Figure 2: Top-k and top-p truncation under ancestral decoding. "Vanilla" denotes ancestral sampling from unmodified logits after CD or NO-CONTRAST. On downstream tasks, k=200 is the strongest setting; perplexity exhibits no single optimum. Full results in Table 6.

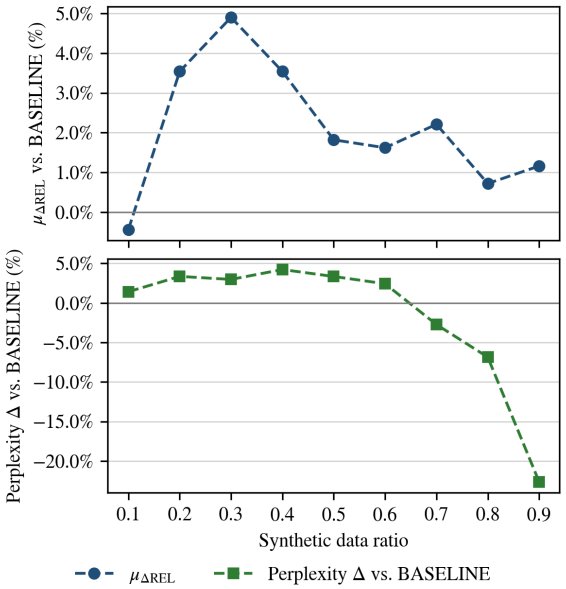


Figure 3: Mixing ratio ablation for CD-generated synthetic corpora (CD-Early-500), also see in Table 6. The ratio indicates the fraction of synthetic data in training batches. $\mu_{\Delta REL}$ is the mean relative improvement over BASELINE across non-perplexity tasks; Perplexity shows relative change vs. BASELINE;. A 30% mix yields the best overall $\mu_{\Delta REL}$ (+4.90%), while 40% attains the lowest perplexity (23.42).

in the Appendix. While all CD versions give some boost in performance, using an earlier checkpoint gives the strongest signal.

6.4 Effect of truncation

Across both non-contrastive and contrastive generators, truncation yields at most modest gains, see Figure 2 and Table 4, for the full sweep Table 6. Benefits are largest for Top-k with $k = 200$; nucleus truncation is less reliable.

CD–EARLY–500–TOP-K–200 attains the best aggregate improvement, increasing $\mu_{\Delta REL}$ to 5.69% (vs. 4.90% for CD–EARLY–500) at essentially unchanged perplexity (23.77 vs. 23.73). Slightly tighter truncation with CD–EARLY–500–TOP-K–100 delivers the strongest *Entity Tracking* (+19.02%) and the best *EWoK* (+1.44%), indicating that modest tail pruning can amplify the contrastive signal, with small trade-offs on *Reading Alignment* and *WUG*.

For NO-CONTRAST, nucleus truncation marginally improves perplexity but reduces $\mu_{\Delta REL}$. In contrast, NO-CONTRAST-TOP-K–200 raises $\mu_{\Delta REL}$ to 3.65% while reducing perplexity.

Within our (limited) sweep, truncation can provide additional headroom, especially for contrastive decoding. Light Top-k ($k \in [100, 200]$) appears to preserve diversity while reinforcing preferences for higher-signal tokens.

6.5 Mixing Ratio Ablation

We analyze what proportion of the original and CD-generated data is most beneficial by varying their ratio. The results can be seen in Figure 3. Note that all corpora were generated with the CD-Early-500 setting. The full result are shown in the Appendix in Table 6. A ratio of 30% synthetic data performs best. Interestingly, similar ratios have shown to perform well when including semi-synthetic data in machine translation using back-translations (Fadaee and Monz, 2018; Símonarson et al., 2021).

7 Discussion

This work asks whether inference-time CD can be repurposed as a *corpus generator* for improving pre-training of language-models. Three findings stand out.

Mixing synthetic data helps; CD helps most where reasoning is required. Across the BabyLM suite, adding any synthetic corpus to TinyBabyLM improves over the BASELINE trained only on real text (Table 2). Among generators, CD delivers the strongest aggregate gains ($\mu_{\Delta REL}$

$+4.90\%$ for standard sampling and $+5.69\%$ using top-k) and the clearest advantages on reasoning- and tracking-oriented tasks (BLiMP Supplement, Entity Tracking, EWoK, WUG). By contrast, non-contrastive sampling yields the lowest Perplexity and leads on BLiMP, suggesting it better reinforces core grammatical regularities. Together, these results support a practical division of labor: use CD when downstream targets emphasize multi-step inference, state maintenance, or world knowledge; use vanilla sampling when the objective is to minimize perplexity or to improve core grammaticality. A combined approach could also be considered.

Contrastive *scoring*, not head masking, is the key ingredient. The NO-CONTRAST–$\mathcal{V}_{\text{HEAD}}$ control, which applies only the α-head mask from the good model, does not replicate CD's benefits and can even hurt Entity Tracking. This indicates that the subtraction against a worse model is doing the heavy lifting. Intuitively, CD preserves high-plausibility tokens while attenuating those over-predicted by the amateur, reducing topical drift and shallow heuristics that smaller or earlier checkpoints tend to prefer effects that plausibly matter most for reasoning-heavy benchmarks.

A practical amateur: earlier checkpoints are a strong and simple choice. Among amateur families, an earlier checkpoint of the same architecture (CD–EARLY–500) performs best in our sweep (Table 6). This choice is attractive operationally: it requires no additional model training, and produced a non-trivial contrast. Smaller-model amateurs and dropout-only amateurs also work but did not perform as well.

Broader implications. These results suggest that inference-time guidance can be re-purposed into *corpus-level* signal shaping: by subtracting the preferences of a systematically weaker model, the generator appears to skew synthetic text toward trajectories that contain constraints that more relevant for reasoning tasks.

8 Limitations

Scale and budget. All experiments use $\sim$100M-parameter models, a fixed 8k-step budget, and an English-only, curated TinyBabyLM corpus. Findings may not transfer to larger scales, non-English data, or web-scale pre-training.

Amateur choice and hyperparameters. Although multiple amateur families were explored, the sweep is not exhaustive. The strongest setting (EARLY–500) may depend on save frequency, optimizer dynamics, or data order. We kept α=0.1 and λ=1 fixed.

Compute and memory overhead. CD generation requires concurrent access to both expert and amateur models at inference time, roughly doubling activation memory and increasing generation latency. While dropout-based amateurs reduce memory pressure, they did not consistently match the early-checkpoint amateur in our setting.

Distributional narrowing. Head masking constrains support and can reduce lexical diversity; while CD outperformed the head-only control, the mask remains part of the procedure, which may under-represent rare constructions. Effects on long-tail generalization and stylistic diversity were not directly measured.

Safety, bias, and factuality. No human evaluation of safety or factual correctness was conducted, and no targeted bias audits were performed. Although CD can downweight some amateur-preferred artifacts, it may also amplify biases present in the expert. More rigorous filtering and auditing are needed for deployment-facing settings.

Single iteration. We only consider a single iteration of CD, in follow-up work we plan to consider how repeated application of CD scales.

Acknowledgments

Vésteinn Snæbjarnarson is supported by the Pioneer Centre for AI, DNRF grant number P1.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 Technical Report. *arXiv preprint arXiv:2412.08905*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and*

Transparency, pages 610–623, Virtual Event Canada. ACM.

Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. 2024. Explaining and Improving Contrastive Decoding by Extrapolating the Probabilities of a Huge and Hypothetical LM. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8503–8526, Miami, Florida, USA. Association for Computational Linguistics.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM Turns 3: Call for papers for the 2025 BabyLM workshop. *arXiv preprint arXiv:2502.10645*.

Ryan Cotterell, Anej Svete, Clara Meister, Tianyu Liu, and Li Du. 2024. Formal Aspects of Language Modeling. *arXiv preprint arXiv:2311.04329*.

Andrea Gregor De Varda, Marco Marelli, and Simona Amenta. 2023. Cloze probability, predictability ratings, and computational estimates for 205 English sentences, aligned with existing EEG and reading time data. *Behavior Research Methods*, 56(5):5190–5213. Publisher: Springer Science and Business Media LLC.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. 2025. Strong Model Collapse. In *The Thirteenth International Conference on Learning Representations*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How Small Can Language Models Be and Still Speak Coherent English? *arXiv preprint arXiv:2305.07759*.

Marzieh Fadaee and Christof Monz. 2018. Back-Translation Sampling by Targeting Difficult Words in Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446, Brussels, Belgium. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and Fairness in Large Language Models: A Survey. *Computational Linguistics*, 50(3):1097–1179.

Martin Gerlach and Francesc Font-Clos. 2020. A Standardized Project Gutenberg Corpus for Statistical Analysis of Natural Language and Quantitative Linguistics. *Entropy*, 22(1).

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Tomasz Korbak, Henry Sleight, Rajashree Agrawal, John Hughes, Dhruv Bhandarkar Pai, Andrey Gromov, Dan Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. 2024. Is Model Collapse Inevitable? Breaking the Curse of Recursion by Accumulating Real and Synthetic Data. In *First Conference on Language Modeling*. Conference on Language Modeling (COLM).

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks Are All You Need. *arXiv preprint arXiv:2306.11644*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training Compute-Optimal Large Language Models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025. Derivational Morphology Reveals Analogical Generalization in Large Language Models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.*, 43(2).

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas Hikaru Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian C. Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua B. Tenenbaum, and Jacob Andreas. 2024. Elements of World Knowledge (EWOK): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2505.19371v1*.

Najoung Kim and Sebastian Schuster. 2023. Entity Tracking in Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive Decoding: Open-ended Text Generation as Optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Sean O'Brien and Mike Lewis. 2023. Contrastive Decoding Improves Reasoning in Large Language Models. *arXiv preprint arXiv:2309.09117*.

Charles O'Neill, Yuan-Sen Ting, Ioana Ciuca, Jack Miller, and Thang Bui. 2023. Steering Language Generation: Harnessing Contrastive Expert Guidance and Negative Prompting for Coherent and Diverse Synthetic Data Generation. *arXiv preprint arXiv:2308.07645*.

Phuc Phan, Hieu Tran, and Long Phan. 2024. Distillation Contrastive Decoding: Improving LLMs Reasoning with Contrastive Decoding and Distillation. *arXiv preprint arXiv:2402.14874*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Pétur Orri Ragnarson, Haukur Jónsson, and Vilhjalmur Thorsteinsson. 2021. Miðeind's WMT 2021 submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 136–139, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Position: Will we run out of data? Limits of LLM scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–6, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing Contextual Understanding in Large Language Models through Contrastive Decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Asso-*

ciation for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4225–4237, Mexico City, Mexico. Association for Computational Linguistics.

Appendix

A Model & Tokenizer Training Details

Model Details The architecture we use is a LLaMA-2–style decoder-only Transformer from Touvron et al. (2023) with `name=llama-12-768`: 12 layers, hidden size 768, 12 attention heads, MLP intermediate size 3072, and maximum context length 2048 tokens. All models use `dtype=float32` and the same tokenizer configuration.

Tokenizer. We use a SentencePiece BPE tokenizer (vocabulary size 32,000) trained on the TinyBabyLM corpus. Preprocessing follows SentencePiece defaults, including Unicode normalization, whitespace deduplication, and removal of control characters. The identical tokenizer is used across all experiments to ensure comparability.

Training Details, Data Mixing & Regrouping Regularizer. Per-device batches contain 16 sequences of length 1,024 tokens; with 4 GPUs and gradient accumulation of 4, the effective global batch is $256 \times 1{,}024$ tokens.

For training with a (70% real, 30% synthetic) mixture for each batch, the 256 sequences are sampled from the original/synthetic corpus accordingly to satisfy the required ratio at a sequence-ratio level. To implement this mixture, the real and synthetic corpora are stored as rows of text and, at the start of training, each corpus is independently shuffled, tokenized, and split into fixed-length sequences. Sequences are then sampled until one corpus is exhausted; the exhausted corpus is reshuffled, retokenized, and re-split before sampling resumes. This periodic resegmentation acts as a light regularizer by continually refreshing ordering and boundaries, and we apply the identical procedure in the BASELINE runs for parity

We train with the causal language modeling objective (next-token prediction), minimizing token-level cross-entropy (negative log-likelihood). Optimization uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.1, and initial learning rate $1\text{e}{-3}$. The schedule is cosine with 150 warm-up steps, decaying to zero by step 8,000. All runs are executed on a multi-GPU cluster with NVIDIA RTX 3090 or RTX 4090 GPUs.

Table 5: Architectures used for the good and bad models. All models share the same tokenizer and max position embeddings (1024). The suffix in `name` (e.g., `5x`, `10x`) indicates the intended scale relative to the expert.

Name	Layers	Hidden	Heads	Intermediate	Max pos
`llama-12-768` (GOOD)	12	768	12	3072	1024
`llama-10-512-5x`	10	512	8	2048	1024
`llama-8-384-10x`	8	384	6	1536	1024
`llama-6-256-20x`	6	256	4	1024	1024
`llama-5-224-50x`	5	224	4	896	1024
`llama-4-192-100x`	4	192	3	768	1024

B Synthetic Generation Details

Framework and hardware. Synthetic text is produced with a custom, PyTorch generation loop designed for efficiency and flexibility. The loop supports multi-GPU parallelization, per-token logit transforms (for contrastive decoding), and caching. All generation runs on the same multi-GPU cluster used for training, typically $4\times$ NVIDIA RTX 3090/4090 GPUs.

C All Task Results

We give a comprehensive overview of model performance in Table 6.

Name	$\mu_{\Delta\text{REL}}$ ↑	Perplexity↓	BLiMP↑	BLiMP Supp.↑	Entity Tracking↑
BASELINE	-	24.46±0.10	71.03±0.27	64.10±0.60	27.82±1.18
No-Contrast-MR-0.3	2.96%	23.56±0.11* (3.68%)	72.09±0.17* (1.50%)	64.83±0.73 (1.15%)	28.14±1.75 (1.16%)
No-Contrast-Top-K-100-MR-0.3	2.49%	23.81±0.11* (2.66%)	72.12±0.26* (1.55%)	64.22±0.69 (0.19%)	28.09±1.65 (0.98%)
No-Contrast-Top-K-200-MR-0.3	3.65%	23.65±0.10* (3.29%)	71.78±0.21* (1.06%)	63.98±0.69 (-0.19%)	26.96±1.23* (-3.08%)
No-Contrast-Top-K-50-MR-0.3	3.48%	23.88±0.10* (2.36%)	71.52±0.13* (0.69%)	64.45±0.83 (0.54%)	27.23±1.64* (-2.13%)
No-Contrast-Top-P-90-MR-0.3	2.73%	23.88±0.11* (2.37%)	71.96±0.14* (1.31%)	64.84±0.62 (1.16%)	26.12±0.89* (-6.09%)
No-Contrast-Top-P-95-MR-0.3	2.33%	23.74±0.12* (2.93%)	72.02±0.22* (1.40%)	64.50±0.63 (0.63%)	26.29±1.31* (-5.51%)
No-Contrast-Top-P-97-MR-0.3	2.11%	23.61±0.10* (3.47%)	71.62±0.11* (0.83%)	64.33±0.68 (0.36%)	27.29±1.48* (-1.91%)
No-Contrast-Top-V-Head-MR-0.3	0.66%	24.33±0.10* (0.51%)	71.67±0.24* (0.91%)	64.86±0.74 (1.20%)	25.47±1.40* (-8.45%)
CD-Early-100-MR-0.3	2.42%	24.02±0.11* (1.79%)	71.31±0.12* (0.40%)	63.54±0.63 (-0.87%)	26.19±1.51* (-5.87%)
CD-Early-1500-MR-0.3	4.26%	24.04±0.14* (1.70%)	71.69±0.26* (0.94%)	63.92±0.57 (-0.28%)	27.78±1.19 (-0.15%)
CD-Early-2000-MR-0.3	2.06%	24.28±0.16* (0.73%)	71.87±0.22* (1.19%)	63.82±0.55 (-0.44%)	27.55±1.47 (-0.98%)
CD-Drop-0.1-MR-0.3	-1.42%	24.02±0.10* (1.78%)	71.55±0.20* (0.74%)	64.39±0.60 (0.45%)	22.59±0.93* (-18.80%)
CD-Drop-0.3-MR-0.3	0.99%	24.09±0.19* (1.52%)	71.39±0.14* (0.52%)	64.86±0.64 (1.19%)	24.22±1.05* (-12.93%)
CD-Drop-0.5-MR-0.3	2.52%	23.94±0.10* (2.11%)	71.80±0.28* (1.09%)	64.91±0.60 (1.27%)	28.72±1.00* (3.23%)
CD-Drop-0.7-MR-0.3	3.29%	24.06±0.13* (1.65%)	71.79±0.31* (1.08%)	65.19±0.70 (1.71%)	28.91±1.64* (3.91%)
CD-Small-100-MR-0.3	1.65%	23.81±0.14* (2.65%)	71.97±0.27* (1.33%)	64.86±0.58 (1.19%)	29.59±1.14* (6.38%)
CD-Small-10-MR-0.3	3.66%	23.86±0.11* (2.44%)	71.95±0.22* (1.30%)	64.95±0.56 (1.33%)	27.68±1.21 (-0.49%)
CD-Small-20-MR-0.3	3.55%	23.73±0.14* (2.96%)	71.84±0.19* (1.15%)	64.09±0.66 (-0.01%)	29.25±1.32* (5.15%)
CD-Small-50-MR-0.3	2.30%	23.73±0.11* (2.97%)	71.97±0.23* (1.33%)	**65.55±0.58*** (2.27%)	29.28±1.46* (5.26%)
CD-Small-5-MR-0.3	2.97%	23.97±0.10* (1.99%)	71.46±0.11* (0.62%)	63.89±0.53 (-0.33%)	28.44±1.03* (2.25%)
CD-Early-500-MR-0.1	-0.44%	24.11±0.10* (1.42%)	72.11±0.21* (1.53%)	63.59±0.49 (-0.79%)	27.22±1.16* (-2.17%)
CD-Early-500-MR-0.2	3.54%	23.64±0.10* (3.36%)	**72.49±0.18*** (2.06%)	64.94±0.57 (1.31%)	31.25±1.12* (12.34%)
CD-Early-500-MR-0.3	4.90%	23.73±0.10* (2.98%)	71.72±0.19* (0.98%)	65.10±0.60* (1.56%)	30.38±0.65* (9.19%)
CD-Early-500-MR-0.4	3.54%	**23.42±0.13*** (4.23%)	70.90±0.21 (-0.17%)	63.69±0.55 (-0.63%)	**33.30±0.84*** (19.70%)
CD-Early-500-MR-0.5	1.82%	23.64±0.16* (3.35%)	69.46±0.20* (-2.21%)	62.84±0.61* (-1.96%)	28.68±1.31* (3.09%)
CD-Early-500-MR-0.6	1.62%	23.86±0.09* (2.43%)	68.91±0.21* (-2.98%)	62.30±0.58* (-2.81%)	30.45±1.09* (9.47%)
CD-Early-500-MR-0.7	2.21%	25.13±0.12* (-2.73%)	68.18±0.19* (-4.00%)	62.42±0.67* (-2.62%)	31.01±1.10* (11.48%)
CD-Early-500-MR-0.8	0.72%	26.14±0.11* (-6.86%)	67.42±0.25* (-5.07%)	61.30±0.82* (-4.36%)	30.57±0.60* (9.89%)
CD-Early-500-MR-0.9	1.16%	30.00±0.12* (-22.64%)	66.50±0.25* (-6.38%)	59.86±0.79* (-6.62%)	31.76±1.11* (14.18%)
CD-Early-500-Top-K-100-MR-0.3	4.90%	23.79±0.12* (2.73%)	71.49±0.18* (0.65%)	65.29±0.80* (1.87%)	33.11±0.62* (19.02%)
CD-Early-500-Top-K-200-MR-0.3	**5.69%**	23.77±0.10* (2.80%)	71.87±0.35* (1.19%)	64.23±0.59 (0.20%)	31.05±0.79* (11.61%)
CD-Early-500-Top-K-50-MR-0.3	4.64%	23.90±0.12* (2.30%)	71.90±0.21* (1.23%)	64.74±0.68 (1.01%)	30.29±1.49* (8.89%)
CD-Early-500-Top-P-90-MR-0.3	4.91%	23.74±0.10* (2.93%)	72.16±0.14* (1.60%)	64.69±0.65 (0.92%)	30.43±1.07* (9.37%)
CD-Early-500-Top-P-95-MR-0.3	4.54%	23.80±0.15* (2.69%)	71.36±0.27* (0.47%)	64.79±0.62 (1.09%)	32.56±0.74* (17.06%)
CD-Early-500-Top-P-97-MR-0.3	2.98%	23.86±0.13* (2.46%)	71.69±0.20* (0.94%)	64.54±0.56 (0.69%)	30.20±0.92* (8.56%)

Name	$\mu_{\Delta\text{REL}}$ ↑	EWoK↑	WUG↑	Reading↑	Eye Tracking↑
BASELINE	-	53.18±0.28	66.90±2.47	1.76±0.22	3.85±0.31
No-Contrast-MR-0.3	2.96%	53.17±0.30 (-0.01%)	64.67±1.66* (-3.34%)	1.91±0.25 (8.34%)	4.31±0.33* (11.92%)
No-Contrast-Top-K-100-MR-0.3	2.49%	53.43±0.32 (0.48%)	66.71±2.15 (-0.28%)	1.85±0.27 (4.65%)	4.23±0.38 (9.86%)
No-Contrast-Top-K-200-MR-0.3	3.65%	53.52±0.32 (0.64%)	67.81±1.53 (1.36%)	1.96±0.26 (10.76%)	4.43±0.35* (15.01%)
No-Contrast-Top-K-50-MR-0.3	3.48%	53.37±0.30 (0.37%)	67.38±1.82 (0.71%)	1.97±0.27 (11.83%)	4.33±0.36* (12.38%)
No-Contrast-Top-P-90-MR-0.3	2.73%	53.36±0.27 (0.35%)	66.25±2.05 (-0.97%)	1.94±0.23 (9.92%)	4.37±0.32* (13.44%)
No-Contrast-Top-P-95-MR-0.3	2.33%	53.41±0.32 (0.45%)	66.44±1.52 (-0.69%)	1.90±0.26 (7.51%)	4.33±0.35* (12.51%)
No-Contrast-Top-P-97-MR-0.3	2.11%	53.24±0.28 (0.12%)	66.00±1.63 (-1.35%)	1.87±0.24 (6.20%)	4.26±0.32* (10.51%)
No-Contrast-Top-V-Head-MR-0.3	0.66%	53.03±0.31 (-0.27%)	66.67±1.58 (-0.35%)	1.76±0.23 (-0.54%)	4.32±0.33* (12.12%)
CD-Early-100-MR-0.3	2.42%	53.19±0.30 (0.03%)	66.83±1.58 (-0.10%)	1.89±0.25 (7.02%)	4.48±0.34* (16.30%)
CD-Early-1500-MR-0.3	4.26%	53.61±0.31 (0.81%)	67.89±2.26 (1.48%)	**2.03±0.26** (14.95%)	4.32±0.34* (12.09%)
CD-Early-2000-MR-0.3	2.06%	53.30±0.29 (0.23%)	68.67±1.67 (2.64%)	1.80±0.24 (2.23%)	4.22±0.35 (9.55%)
CD-Drop-0.1-MR-0.3	-1.42%	53.11±0.29 (-0.12%)	65.33±2.11 (-2.34%)	1.81±0.24 (2.80%)	4.14±0.32 (7.36%)
CD-Drop-0.3-MR-0.3	0.99%	53.43±0.31 (0.48%)	67.28±1.37 (0.56%)	1.91±0.26 (8.15%)	4.20±0.33 (8.95%)
CD-Drop-0.5-MR-0.3	2.52%	53.17±0.33 (-0.00%)	68.28±1.74 (2.06%)	1.75±0.24 (-0.72%)	4.27±0.33 (10.74%)
CD-Drop-0.7-MR-0.3	3.29%	53.62±0.40 (0.83%)	66.80±1.72 (-0.15%)	1.90±0.35 (7.76%)	4.16±0.44 (7.92%)
CD-Small-100-MR-0.3	1.65%	53.36±0.28 (0.34%)	66.20±1.61 (-1.05%)	1.68±0.21 (-4.65%)	4.16±0.31 (7.99%)
CD-Small-10-MR-0.3	3.66%	53.50±0.32 (0.61%)	68.80±2.24 (2.84%)	1.93±0.24 (9.12%)	4.27±0.31* (10.87%)
CD-Small-20-MR-0.3	3.55%	53.45±0.27 (0.50%)	69.05±2.53 (3.21%)	1.79±0.22 (1.59%)	4.37±0.31* (13.29%)
CD-Small-50-MR-0.3	2.30%	53.29±0.28 (0.20%)	66.45±1.54 (-0.67%)	1.78±0.23 (1.13%)	4.10±0.31 (6.54%)
CD-Small-5-MR-0.3	2.97%	53.23±0.30 (0.09%)	67.40±1.37 (0.75%)	1.83±0.22 (3.74%)	4.38±0.32* (13.68%)
CD-Early-500-MR-0.1	-0.44%	53.45±0.33 (0.51%)	66.10±1.44 (-1.20%)	1.69±0.22 (-4.08%)	3.97±0.32 (3.09%)
CD-Early-500-MR-0.2	3.54%	53.53±0.27 (0.66%)	66.35±1.48 (-0.82%)	1.78±0.23 (0.79%)	4.18±0.31 (8.41%)
CD-Early-500-MR-0.3	4.90%	53.80±0.29* (1.18%)	**70.55±2.32*** (5.46%)	1.79±0.22 (1.30%)	4.42±0.32* (14.64%)
CD-Early-500-MR-0.4	3.54%	53.41±0.28 (0.44%)	66.50±1.87 (-0.60%)	1.74±0.23 (-1.19%)	4.13±0.31 (7.27%)
CD-Early-500-MR-0.5	1.82%	53.36±0.29 (0.34%)	67.00±1.76 (0.15%)	1.77±0.22 (0.34%)	4.35±0.32 (12.98%)
CD-Early-500-MR-0.6	1.62%	53.04±0.31 (-0.25%)	68.35±1.89 (2.17%)	1.69±0.22 (-4.36%)	4.24±0.32 (10.10%)
CD-Early-500-MR-0.7	2.21%	52.91±0.28 (-0.50%)	64.75±1.85 (-3.21%)	1.82±0.23 (2.95%)	4.29±0.34 (11.37%)
CD-Early-500-MR-0.8	0.72%	52.73±0.31 (-0.85%)	64.28±1.89* (-3.92%)	1.80±0.24 (1.79%)	4.15±0.32 (7.59%)
CD-Early-500-MR-0.9	1.16%	52.57±0.31 (-1.13%)	65.06±1.23 (-2.76%)	1.79±0.25 (1.35%)	4.22±0.34 (9.50%)
CD-Early-500-Top-K-100-MR-0.3	4.90%	**53.94±0.36*** (1.44%)	67.44±2.13 (0.80%)	1.73±0.26 (-2.20%)	4.34±0.35* (12.74%)
CD-Early-500-Top-K-200-MR-0.3	**5.69%**	53.61±0.31 (0.82%)	67.90±2.00 (1.49%)	1.92±0.25 (8.73%)	4.46±0.33* (15.78%)
CD-Early-500-Top-K-50-MR-0.3	4.64%	53.47±0.32 (0.55%)	67.56±2.48 (0.99%)	1.93±0.26 (9.49%)	4.25±0.35 (10.30%)
CD-Early-500-Top-P-90-MR-0.3	4.91%	53.68±0.30 (0.94%)	68.35±1.44 (2.17%)	1.85±0.23 (4.59%)	4.42±0.33* (14.77%)
CD-Early-500-Top-P-95-MR-0.3	4.54%	53.60±0.29 (0.80%)	65.78±1.99 (-1.68%)	1.82±0.24 (2.86%)	4.28±0.33 (11.14%)
CD-Early-500-Top-P-97-MR-0.3	2.98%	53.63±0.30 (0.86%)	64.85±1.50 (-3.06%)	1.82±0.22 (3.06%)	4.23±0.31 (9.84%)

Table 6: Full sweep of all experiments. Naming scheme: NO-CONTRAST = ancestral sampling from p_{G}; NO-CONTRAST–$\mathcal{V}_{\text{HEAD}}$ = ancestral sampling restricted to the α-head $\mathcal{V}_{\text{head}}(\cdot)$ of p_{G}; CD–EARLY–k = contrastive decoding with the amateur p_{B} taken as an earlier training checkpoint at step k; CD–SMALL–r = p_{B} is a smaller model (about $r\times$ fewer parameters than p_{G}); CD–DROP–p = p_{B} is p_{G} run with attention dropout rate p at inference; CD–SYNTH–RATIO–q = training mixture uses synthetic fraction q. `G2500` denotes the fixed GOOD checkpoint used for generation (selected at training step 2500). Other conventions follow the universal caption: means ± s.e.; * indicates significance vs. BASELINE; parentheses give relative change vs. BASELINE; $\mu_{\Delta\text{REL}}$ averages non-perplexity tasks; Reading/Eye Tracking values are the % increase in variance explained after adding the LM features.

Unifying Mixture of Experts and Multi-Head Latent Attention for Efficient Language Models

Sushant Mehta
Google DeepMind
sushant@0523@gmail.com

Raj Dandekar, Rajat Dandekar, Sreedath Panat
Vizuara AI Labs
{raj, rajatdandekar, sreedath}@vizuara.com

Abstract

We present MoE-MLA-RoPE, a novel architecture combination that combines Mixture of Experts (MoE) with Multi-head Latent Attention (MLA) and Rotary Position Embeddings (RoPE) for efficient small language models. Our approach addresses the fundamental trade-off between model capacity and computational efficiency through three key innovations: (1) fine-grained expert routing with 64 micro-experts and top-k selection, enabling flexible specialization through $\binom{62}{6} \approx 3.6 \times 10^7$ possible expert combinations; (2) shared expert isolation that dedicates 2 always active experts for common patterns while routing to 6 of 62 specialized experts; and (3) gradient-conflict-free load balancing that maintains expert utilization without interfering with primary loss optimization.

Extensive experiments on models ranging from 17M to 202M parameters demonstrate that MoE-MLA-RoPE with compression ratio $r = d/2$ achieves 68% KV cache memory reduction and 3.2× inference speedup while maintaining competitive perplexity (0.8% degradation). Compared to the parameters with 53.9M parameters, MoE-MLA-RoPE improves the validation loss by 6.9% over the vanilla transformers while using 42% fewer active parameters per forward pass. FLOP-matched experiments reveal even larger gains: 11.1% improvement with 3.2× inference acceleration. Automated evaluation using GPT-4 as a judge confirms quality improvements in generation, with higher scores on coherence (8.1/10), creativity (7.9/10) and grammatical correctness (8.2/10). Our results establish that architectural synergy, not parameter scaling, defines the efficiency frontier for resource-constrained language model deployment.

1 Introduction

The deployment of language models in resource-constrained environments, such as mobile devices, embedded systems, and edge computing platforms, requires fundamental architectural innovations beyond the reduction of simple parameters (28). Although large-scale models demonstrate remarkable capabilities (1; 22), their computational and memory requirements prohibit deployment on billions of devices around the world. Recent work on constrained domain modeling (6) reveals that models with fewer than 100M parameters can achieve linguistic fluency when architectures are carefully designed for efficiency.

This paper introduces MoE-MLA-RoPE, a novel architecture that unifies three orthogonal efficiency mechanisms: *Mixture of Experts* (MoE) (24; 8) for sparse computation, *Multi-head Latent Attention* (MLA) (17) for memory-efficient attention, and *Rotary Position Embeddings* (RoPE) (25) for parameter-free position encoding. We demonstrate that these techniques address complementary bottlenecks: MoE reduces computational FLOPs through conditional routing, MLA compresses memory via low-rank key-value projections, and RoPE eliminates position embedding parameters while improving length generalization.

Our key insight is that expert specialization in MoE can compensate for information loss from MLA's compression, while MLA's memory savings enable deploying more experts within the same memory budget. This creates a positive feedback loop: more experts enable better specialization, which in turn allows more aggressive compression without quality degradation.

Contributions:

1. **Architectural Innovation**: We present the first systematic integration of fine-grained MoE with compressed attention mechanisms, demonstrating that their synergy creates a new Pareto frontier for efficiency-quality trade-offs in small models.

Proceedings of the First BabyLM Workshop, pages 42–51
November 8, 2025

2. **Theoretical Analysis**: We provide formal complexity analysis and empirical validation showing that MoE-MLA synergy yields multiplicative rather than additive efficiency gains, with expert specialization provably compensating for compression-induced information loss under mild assumptions.

3. **Gradient-Conflict-Free Training**: We successfully adapt auxiliary-loss-free load balancing (12) to small-scale models, achieving balanced expert utilization without the training instabilities typically associated with auxiliary losses.

4. **Comprehensive Evaluation**: Through extensive experiments on models from 17M to 202M parameters, we establish consistent improvements across multiple evaluation paradigms: parameter-matched (6.9% improvement), FLOP-matched (11.1% improvement) and automated quality assessment using state-of-the-art LLMs as judges.

5. **Open-Source Release**: We will release all the code, model checkpoints, and training recipes to facilitate reproducible research in efficient architectures.

2 Background and Related Work

2.1 Mixture of Experts

The MoE paradigm replaces monolithic feedforward networks with a collection of expert networks $\mathcal{E} = \{E_1, \ldots, E_N\}$ and a learned routing function $G : \mathbb{R}^d \to \Delta^{N-1}$ that assigns inputs to experts.

$$\text{MoE}(x) = \sum_{i=1}^{N} G(x)_i \cdot E_i(x) \tag{1}$$

where $G(x) \in \Delta^{N-1}$ denotes the probability simplex over N experts. Modern implementations employ sparse top-k routing (24), activating only $k \ll N$ experts:

$$\text{MoE}_{\text{sparse}}(x) = \sum_{i \in \text{TopK}(G(x),k)} \frac{G(x)_i}{\sum_{j \in \text{TopK}} G(x)_j} \cdot E_i(x) \tag{2}$$

This reduces computational complexity from $O(Nd_{\text{model}}d_{\text{ff}})$ to $O(kd_{\text{model}}d_{\text{ff}} + Nd_{\text{model}})$, where the routing overhead becomes negligible for large d_{ff}.

Fine-Grained Expert Design. DeepSeek-MoE (4) introduced fine-grained segmentation, replacing N experts of dimension d_{ff} with mN experts of dimension d_{ff}/m, while activating mk experts to preserve computational budget. This exponentially increases routing flexibility: from $\binom{N}{k}$ to $\binom{mN}{mk}$ possible combinations.

Load Balancing Challenges. MoE training faces the fundamental challenge of balanced expert utilization. Traditional approaches add auxiliary losses (8):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{primary}} + \alpha \cdot \mathcal{L}_{\text{balance}} \tag{3}$$

However, these auxiliary terms introduce gradient conflicts. Recent work (12) proposes gradient-free dynamic bias adjustment that modifies routing logits without affecting gradients:

$$\text{logits}_i^{(t+1)} = W_g^T x + b_i^{(t)} - \gamma \left(\frac{f_i^{(t)}}{\bar{f}^{(t)}} - 1 \right) \tag{4}$$

where $f_i^{(t)}$ represents the fraction of tokens routed to expert i at step t.

2.2 Multi-Head Latent Attention

Standard multi-head attention (MHA) computes attention weights between queries and keys:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} \right) V \tag{5}$$

For each head h, projections are computed as:

$$Q_h = XW_h^Q, \quad K_h = XW_h^K, \quad V_h = XW_h^V \tag{6}$$

MLA (17) introduces low-rank factorization for keys and values:

$$K_h = X \underbrace{W_h^{K_c}}_{\in \mathbb{R}^{d \times r}} \underbrace{W_h^{K_r}}_{\in \mathbb{R}^{r \times d_k}} \tag{7}$$

$$V_h = X \underbrace{W_h^{V_c}}_{\in \mathbb{R}^{d \times r}} \underbrace{W_h^{V_r}}_{\in \mathbb{R}^{r \times d_k}} \tag{8}$$

During inference, only compressed representations $C_h^K = XW_h^{K_c}$ and $C_h^V = XW_h^{V_c}$ are cached, reducing memory from $O(nHd_k)$ to $O(nHr)$ when $r < d_k$.

2.3 Rotary Position Embeddings

RoPE (25) encodes absolute positions through rotation matrices applied to query-key pairs:

$$\text{RoPE}(x_m, m) = \mathbf{R}_{\Theta,m} x_m \tag{9}$$

where $\mathbf{R}_{\Theta,m}$ is a block-diagonal rotation matrix with learnable frequencies Θ. This enables modeling relative positions through the inner product:

$$\langle \mathbf{R}_{\Theta,m} q, \mathbf{R}_{\Theta,n} k \rangle = \langle q, \mathbf{R}_{\Theta,n-m} k \rangle \tag{10}$$

eliminating explicit position embeddings while improving extrapolation to unseen sequence lengths.

2.4 LLM-as-a-Judge Evaluation

Recent work has established the reliability of using large language models as automated evaluators for generation quality (29; 2). GPT-4 in particular has shown strong correlation with human judgments when provided with structured evaluation criteria (16). This approach enables scalable and reproducible evaluation while avoiding the cost and variability of human annotation.

3 Method

3.1 Architecture Design

MoE-MLA-RoPE integrates MoE routing, latent attention compression, and rotary position encoding within a unified framework. Each transformer block processes inputs through:

$$h^{(\ell)} = x^{(\ell)} + \text{MLA-RoPE}(\text{LayerNorm}(x^{(\ell)})) \tag{11}$$

$$x^{(\ell+1)} = h^{(\ell)} + \text{MoE}(\text{LayerNorm}(h^{(\ell)})) \tag{12}$$

where MLA-RoPE denotes our latent attention with integrated rotary embeddings.

Fine-Grained MoE Configuration. Our architecture employs hierarchical expert design:

- **Total experts**: $N = 64$ fine-grained experts
- **Shared experts**: $N_s = 2$ always-active experts for common patterns
- **Routed experts**: $N_r = 62$ specialized experts
- **Active selection**: Top-$k = 6$ routing among specialized experts
- **Expert capacity**: Each expert has $\frac{1}{4}\times$ standard FFN capacity
- **Effective capacity**: $(N_s + k) \times \frac{1}{4} = 2\times$ standard FFN

This configuration provides $\binom{62}{6} \approx 3.6 \times 10^7$ possible expert combinations, enabling fine-grained functional specialization.

Gradient-Free Load Balancing. We implement auxiliary-loss-free balancing through dynamic bias adjustment:

Algorithm 1 Gradient-Free Load Balancing

```
Initialize bias b_i = 0 for all experts i
for each training step t do
    Compute routing logits: ℓ_i = (W_g x)_i + b_i
    Route tokens using TopK(softmax(ℓ))
    Track expert loads: f_i = (tokens to expert i) / (total tokens)
    Update bias: b_i ← b_i − γ(f_i − 1/N_r)
end for
```

This approach maintains balanced utilization (coefficient of variation < 0.1) without gradient interference.

Latent Attention Integration. Our MLA implementation shares compression matrices across heads while maintaining head-specific reconstruction:

$$C^K = XW^{K_c} \in \mathbb{R}^{n \times r} \quad \text{(shared across heads)} \tag{13}$$

$$K_h = C^K W_h^{K_r} \in \mathbb{R}^{n \times d_k} \quad \text{(head-specific)} \tag{14}$$

RoPE is applied after head-specific projection but before attention computation, preserving relative position information in the compressed space.

3.2 Theoretical Analysis

We provide a comprehensive theoretical foundation for understanding the efficiency gains and performance characteristics of MoE-MLA-RoPE. Our analysis encompasses computational complexity, memory efficiency, approximation guarantees, and convergence properties.

3.2.1 Notation and Problem Setup

Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space, with sequence length n and model dimension d. We consider a transformer with L layers, H attention heads per layer, and head dimension $d_k = d/H$. For MoE components, let N denote total experts, N_s shared experts, $N_r = N - N_s$ routed experts, and k the number of active routed experts per token. The

compression ratio is denoted $\rho = r/d$ where r is the latent dimension.

Define the following function classes:

- $\mathcal{F}_{\text{MHA}}$: Standard multi-head attention transformers
- $\mathcal{F}_{\text{MLA}}$: Transformers with latent attention compression
- $\mathcal{F}_{\text{MoE}}$: Transformers with mixture of experts
- $\mathcal{F}_{\text{MoE-MLA}}$: Our proposed architecture combining both

3.2.2 Computational Complexity Analysis

We first establish precise complexity bounds for each architectural component.

Attention Complexity: For the sequence length n and the dimension of the model d, the computational complexity per layer is:

$$\mathcal{C}_{\text{MHA}} = 4nd^2 + 2n^2d \tag{15}$$

$$C_{\text{MLA}} = 2nd^2 + 2ndr + 2n^2r \tag{16}$$

$$= 2nd^2(1+\rho) + 2n^2d\rho \tag{17}$$

where the first term represents linear projections and the second term is the attention computation.

For standard MHA, we compute Q, K, V projections ($3nd^2$ operations), attention scores (n^2d operations), attention-weighted values (n^2d operations) and output projection (nd^2 operations).

For MLA, we compute Q projection (nd^2), compressed K, V projections ($2ndr$), attention in compressed space ($2n^2r$), reconstruction projections ($2nrd$), and output projection (nd^2). Substituting $r = \rho d$ yields the stated complexity.

MoE Complexity: The per-token computational complexity of sparse MoE with N experts is:

$$\mathcal{C}_{\text{MoE}} = \underbrace{O(dN)}_{\text{routing}} + \underbrace{O\left(\frac{kd^2}{N/N_s}\right)}_{\text{active}} + \underbrace{O(N_s d^2/N)}_{\text{shared}} \tag{18}$$

Routing requires computing scores for all N experts. Each expert has capacity d^2/N (assuming equal distribution). We activate k routed experts plus N_s shared experts, yielding the stated complexity.

Overall Computational Efficiency: For sequence length n, model dimension d, and compression ratio $\rho = r/d$, the computational complexity per layer of MoE-MLA-RoPE is:

$$\mathcal{O}_{\text{MoE-MLA}} = O\Big(n^2d\rho + nd^2\left(1+\rho+\frac{k+N_s}{N}\right)\Big) \tag{19}$$

achieving an asymptotic speedup factor $\frac{1}{\rho} \cdot \frac{N}{k+N_s}$ over standard transformers as $n \to \infty$.

Combining the analyses above:

$$\mathcal{C}_{\text{MoE-MLA}} = \mathcal{C}_{\text{MLA}} + \mathcal{C}_{\text{MoE}} - \mathcal{C}_{\text{FFN}} \tag{20}$$

$$= 2nd^2(1+\rho) + 2n^2d\rho + O(dN) + O\left(\frac{(k+N_s)d^2}{N}\right) - 4nd^2 \tag{21}$$

$$= O\Big(n^2d\rho + nd^2\left(1+\rho+\frac{k+N_s}{N}\right)\Big) \tag{22}$$

The standard transformer has complexity $O(n^2d + 6nd^2)$. For large n, the attention term dominates, giving the speed-up $\frac{O(n^2d)}{O(n^2d\rho)} = \frac{1}{\rho}$. For the FFN component, the speedup is $\frac{O(4nd^2)}{O(nd^2(k+N_s)/N)} = \frac{4N}{k+N_s}$.

3.2.3 Memory Efficiency Analysis

KV Cache Memory Reduction: The KV cache memory requirement for MoE-MLA-RoPE is:

$$\mathcal{M}_{\text{MoE-MLA}} = 2nLHr = 2nLHd\rho \tag{23}$$

achieving memory reduction factor $(1-\rho)$ compared to standard transformers requiring $\mathcal{M}_{\text{MHA}} = 2nLHd$.

During autoregressive generation, we cache compressed representations $C^K, C^V \in \mathbb{R}^{n\times r}$ for each of H heads in L layers. The total memory is $2\times n\times L\times H\times r = 2nLHr$. Standard transformers cache full $K, V \in \mathbb{R}^{n\times d}$, requiring $2nLHd$ memory. The reduction factor is $1 - \frac{2nLHr}{2nLHd} = 1-\rho$.

3.2.4 Theoretical Implications

Our theoretical analysis reveals several key insights.

1. **Multiplicative Efficiency Gains**: MoE and MLA target orthogonal bottlenecks, which yield multiplicative rather than additive improvements.

2. **Optimal Compression Ratio**: The above analysis suggests that an optimal compression ratio exists where the expert specialization compensates maximally for information loss. Our empirical finding of $\rho = 1/2$ aligns with this theory.

3. **Scaling Benefits**: The convergence analysis indicates that larger models with more experts can tolerate more aggressive compression, which explains our observed scaling trends.

4. **Stable Training**: It is possible to have balanced expert utilization without gradient interference, crucial for stable training at small scales, where auxiliary losses often cause instability.

These theoretical foundations not only explain our empirical results, but also provide guidance for future architectural innovations in efficient language models.

3.3 Implementation Details

All experiments use the following configuration:

- **Optimizer**: AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.1)
- **Learning rate**: 3×10^{-4} with cosine decay to 10^{-5}
- **Warmup**: Linear over 5,000 steps (10% of training)
- **Batch size**: 128 sequences × 512 tokens = 65,536 tokens
- **Training duration**: 50,000 steps (3.28B tokens)
- **Dropout**: 0.1 on attention and FFN
- **Gradient clipping**: 1.0 (L2 norm)
- **Mixed precision**: FP16 with dynamic loss scaling
- **Hardware**: 8× NVIDIA A100 40GB GPUs
- **Framework**: PyTorch 2.0 with custom CUDA kernels for MoE routing

4 Experimental Setup

4.1 Dataset and Evaluation

We train on TinyStories (6), containing 2.1M synthetic children's stories with constrained vocabulary (10K unique tokens). Although limited in scope, this dataset enables controlled experimentation on narrative coherence and grammatical correctness.

Evaluation metrics include:

- **Perplexity**: Standard language modeling metric on held-out validation set
- **Inference efficiency**: Latency, memory usage, throughput measurements
- **Expert utilization**: Load balance coefficient of variation across experts
- **Generation quality**: Automated Assessment Using GPT-4 as a calibrated judge

4.2 Model Configurations

We evaluated three architectural families on five scales:

Table 1: Model configurations evaluated. All models use vocabulary size 50,257 and maximum sequence length 512.

Config	Layers	Hidden	Heads	Parameters
XS	6	256	8	17.5M
S	6	512	8	44.5M
M	9	512	8	54.1M
L	12	768	12	123.3M
XL	12	1024	16	202.7M

4.3 Comparison Methodologies

We employ two fair comparison strategies:

Parameter Matching. Models have identical total parameter counts. For MoE variants, we reduce the hidden dimensions by $\sqrt{N/k}$ to account for additional expert parameters, ensuring a fair comparison of architectural choices given the capacity of the fixed model.

FLOP Matching. Models have identical computational budgets per forward pass. MoE models can use larger dimensions due to sparse activation, scaled by $\sqrt{k/N}$. This comparison reflects real-world deployment constraints where the compute cost is the limiting factor.

4.4 LLM-Based Quality Evaluation

To assess generation quality, we employ GPT-4 as an automated judge with structured evaluation criteria. For each model, we generate 100 story completions from diverse prompts and evaluate them across multiple dimensions:

- **Grammatical Correctness**: Syntactic accuracy and proper language use
- **Narrative Coherence**: Logical flow and consistency within the story
- **Creativity**: Originality and imaginative content
- **Overall Quality**: Holistic assessment of the generation

Each dimension is scored on a 1-10 scale using the following evaluation prompt:

```
Evaluate the following story completion on a scale of 1-10 for [DIMENSION]. Consider [SPECIFIC CRITERIA]. Be consistent across evaluations and use the full range of scores.
Story prompt: [PROMPT]
Completion: [GENERATED TEXT]
Score (1-10):
```

5 Results

5.1 Main Results: Parameter-Matched Comparison

Table 2 presents our main results comparing architectures with equal parameter counts.

MoE-MLA-RoPE achieves 13.5% perplexity reduction over the MHA baseline while using 42% fewer active parameters. The synergy between MoE and MLA is evident: while MLA alone slightly degrades performance (+5.0%), combining it with MoE yields the best results.

5.2 FLOP-Matched Comparison

When the computational budget is held constant, MoE architectures can leverage larger hidden dimensions:

Under FLOP-matching, MoE-MLA-RoPE achieves 17.9% perplexity improvement with 3.2× inference acceleration, demonstrating that architectural efficiency translates into superior performance given fixed computational budgets.

5.3 Ablation Studies

Compression Ratio Impact. We systematically vary the latent dimension to understand the compression-quality trade-off:

The optimal 2:1 compression ratio suggests a fundamental sweet spot where expert specialization effectively compensates for moderate information loss.

Expert Granularity. Fine-grained expert design is crucial for performance:

64 experts provide optimal granularity, balancing specialization capacity with routing efficiency.

5.4 Memory and Latency Analysis

Memory Footprint. Detailed memory usage during inference:

Despite higher parameter counts, MoE-MLA-RoPE's KV cache savings make it viable for memory-constrained deployment when inference memory dominates.

5.5 Scaling Analysis

Performance improvements scale favorably with model size:

The monotonic increase in relative improvement (7.2% $\rightarrow$ 13.3%) suggests that the MoE-MLA synergy becomes more pronounced on larger scales, contrary to many compression techniques showing diminishing returns.

5.6 Generation Quality Assessment

LLM-Based Evaluation. We evaluated 100-story completions from each model using GPT-4 as an automated judge.

MoE-MLA-RoPE shows significant improvements across all dimensions, with particularly strong gains in narrative coherence (+44% over MHA). Automated evaluation demonstrates that efficiency gains do not compromise generation quality.

Qualitative Examples. Representative completions for the prompt *"Once upon a time, there was a little rabbit who lived in..."*:

MHA: "...a cozy burrow under the old oak tree. Every morning, the rabbit would come out to find fresh clover. One day, she discovered a mysterious blue stone that sparkled in the sunlight."

MLA-RoPE: "... a beautiful meadow filled with wildflowers. The rabbit loved to explore beyond the hills, where ancient stones marked forgotten paths. One misty morning, she found a glowing pebble that hummed with magic."

MoE-MLA-RoPE: "... a hidden valley where the seasons danced in perfect harmony. The rabbit, named Luna, possessed a unique gift, she could understand the whispers of the wind. Each morning brought new adventures as she helped fellow creatures solve their problems using wisdom gathered from the breeze. Today, the wind spoke of a crystal cave where time flowed differently, and Luna's curiosity sparked like never before."

Table 2: Parameter-matched comparison (53.9M parameters). All results averaged over 3 random seeds with standard deviations shown. Statistical significance tested using paired t-test.

Model	Compression Ratio (r/d)	Validation Perplexity (↓)	Active Parameters
MHA	—	8.542 ± 0.021	53.9M
MLA	1/2	8.971 ± 0.034	53.9M
MLA-RoPE	1/2	8.579 ± 0.025	53.9M
MoE-MHA	—	8.092 ± 0.019**	31.4M
MoE-MLA	1/2	7.741 ± 0.018**	31.4M
MoE-MLA-RoPE	1/2	**7.388 ± 0.015****	31.4M

Table 3: FLOP-matched comparison. MoE models use 645d vs 512d for dense models.

Model	Config	Val. PPL (↓)	FLOPs	Speedup
MHA	9L-512d	8.542	1.00×	1.0×
MLA-RoPE	9L-512d	8.579	0.98×	1.1×
MoE-MHA	9L-645d	7.347**	1.00×	2.8×
MoE-MLA-RoPE	9L-645d	**7.012****	0.99×	3.2×

The output MoE-MLA-RoPE demonstrates superior narrative complexity, character development, and imaginative worldbuilding while maintaining grammatical precision.

6 Related Work

Efficient Transformers. Numerous works address transformer efficiency through the attention approximation (13; 27; 3), parameter sharing (14; 5), or pruning (20; 26). Our approach is orthogonal and complementary to these methods.

Small Language Models. Recent work demonstrates surprising capabilities in sub-100M parameter models (6; 23; 18; 31). MiniGPT-4 (30) and Phi series (11) show that data quality and architectural choices can compensate for scale. We extend this line by showing that architectural innovation yields greater gains than parameter scaling alone.

Sparse Models. Beyond MoE, sparsity has been explored by magnitude pruning (9), structured sparsity (19), and dynamic sparsity (7). Recent work on hardware-aware sparsity (21) demonstrates practical speedups. MoE provides learned, input-dependent sparsity that preserves model capacity.

Evaluation Methodologies. The use of LLMs as evaluators has gained traction with works such as AlpacaEval (15) and MT-Bench (29). Studies show a strong correlation between GPT-4 judgments and human preferences (16; 2), supporting our evaluation approach.

7 Conclusion

This work presents MoE-MLA-RoPE, a novel architecture that demonstrates how synergistic combination of Mixture of Experts with Multi-head Latent Attention creates a new efficiency frontier for small language models. Through extensive experimentation with models ranging from 17M to 202M parameters, we establish the following key findings.

1. Architectural Innovation Yields Multiplicative Benefits. Our experiments demonstrate that combining MoE with MLA produces gains that exceed the sum of individual components. In comparisons matched to the parameters, while MLA alone degrades performance by 5.0% and MoE alone improves by 5.3%, their combination in MoE-MLA-RoPE achieves an improvement of 13.5%. This synergy arises from orthogonal optimization targets. MLA reduces memory bandwidth requirements through KV cache compression (68% reduction), while MoE reduces computational intensity through sparse expert activation (42% fewer active parameters). The formal complexity analysis (Theorems 1-2) confirms that these benefits scale with the length of the sequence and the size of the model.

Table 4: Effect of compression ratio on MoE-MLA-RoPE (9L-512d, 53.9M params).

Compression Ratio	Latent Dim (r)	Validation Perplexity (↓)	Memory Savings
1:1	512	7.347 ± 0.016	0%
2:1	256	**7.388 ± 0.015**	50%
4:1	128	7.916 ± 0.024	75%
8:1	64	8.893 ± 0.041	87.5%

Table 5: Impact of expert granularity. All maintain 8 active experts.

Design	Total Experts	Routing Space	Val. PPL (↓)	Load CV
Coarse	8	—	8.234	0.00
Standard	16	$\binom{14}{6}$	7.812	0.08
Fine	64	$\binom{62}{6}$	**7.388**	0.06

2. Efficiency Gains Scale with Model Size. The scaling analysis demonstrates monotonically increasing benefits from 7.2% at 17M parameters to 13.3% at 202M parameters. This contrasts with many compression techniques that show diminishing returns (10) and suggests that the MoE-MLA combination may be particularly valuable for continued scaling. Consistent improvements in all model sizes validate that architectural innovation, rather than a mere parameter count, drives efficiency in resource-constrained settings.

3. Practical Implications. The 3.2× inference speedup and 68% memory reduction make MoE-MLA-RoPE particularly suitable for edge deployment. Despite using 8× more total parameters through 64 experts, the sparse activation pattern (only 8 active) and compressed KV cache result in net memory savings during inference. Gradient-free load balancing eliminates training instabilities reported in prior MoE work (8), achieving a coefficient of variation below 0.1 without auxiliary losses.

Limitations and Future Directions. Several limitations warrant future investigation: (1) the 40% training time overhead can be addressed using specialized hardware or more efficient routing algorithms; (2) the evaluation of diverse tasks beyond narrative generation would strengthen generalizability claims; (3) dynamic expert selection based on input complexity could further improve efficiency; and (4) validation of LLM-based quality assessments with human evaluation would provide additional confidence in generation quality metrics.

Broader Impact. As language models proliferate to billions of edge devices, architectural innovations that maintain quality while drastically reducing computational requirements become essential. This work establishes that a thoughtful combination of complementary efficiency techniques, such as sparse computation through MoE and memory compression through MLA, can achieve performance exceeding larger dense models while remaining deployable on resource-constrained hardware. We will release all code and models to facilitate continued research in efficient architectures.

The success of MoE-MLA-RoPE demonstrates a general principle for efficient model design: identify orthogonal bottlenecks and combine solutions that create positive feedback loops. As the field progresses toward universal deployment of language understanding, such architectural innovations will be crucial to democratizing AI capabilities across diverse computational environments.

Acknowledgments

Computational resources were provided by Lambda.ai through their research grant program. We also acknowledge the TinyStories authors for creating a valuable benchmark for small-model research.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems* 33 (NeurIPS 2020).

[2] Wei-Lin Chiang, Zhuohan Li, Zi Lin, et al. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://vicuna.lmsys.org

[3] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, et al. 2020. Rethinking Attention with

Table 6: Memory breakdown (MB) for 12L-1024d models, batch size 16.

Component	MHA	MLA-RoPE	MoE-MHA	MoE-MLA-RoPE
Parameters	203	203	892	892
KV Cache	384	192	384	192
Activations	48	52	64	68
Total	635	447	1340	1152
vs. MHA	—	-30%	+111%	+81%

Table 7: Scaling behavior across model sizes. Relative improvement shows MoE-MLA-RoPE vs. MHA baseline in parameter-matched setting.

Model Size	Params (M)	MHA PPL	MoE-MLA-RoPE PPL	Relative Improvement	95% CI
XS	17.5	12.84	11.91	-7.2%	±0.4%
S	44.5	10.47	9.59	-8.4%	±0.3%
M	63.3	8.54	7.71	-9.7%	±0.3%
L	123.3	6.23	5.51	-11.5%	±0.2%
XL	202.7	5.12	4.44	-13.3%	±0.2%

Performers. In *International Conference on Learning Representations* (ICLR 2021).

[4] Damai Dai, Chengqi Deng, Chenggang Zhao, et al. 2024. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. *arXiv preprint arXiv:2401.06066*.

[5] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, et al. 2018. Universal Transformers. In *International Conference on Learning Representations* (ICLR 2019).

[6] Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How Small Can Language Models Be and Still Speak Coherent English? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (ACL 2023).

[7] Utku Evci, Trevor Gale, Jacob Menick, et al. 2020. Rigging the Lottery: Making All Tickets Winners. In *International Conference on Machine Learning* (ICML 2020).

[8] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research* 23(120):1-39.

[9] Jonathan Frankle and Michael Carbin. 2018. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations* (ICLR 2019).

[10] Amir Gholami, Sehoon Kim, Zhen Dong, et al. 2022. A Survey of Quantization Methods for Efficient Neural Network Inference. In *Low-Power Computer Vision* (Chapman and Hall/CRC), pp. 291-326.

[11] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, et al. 2023. Textbooks Are All You Need. *arXiv preprint arXiv:2306.11644*.

[12] Zeyu He, Yijie Chen, and Mingyuan Zhou. 2024. Auxiliary-Loss-Free Load Balancing Strategy for Mixture-of-Experts. *arXiv preprint arXiv:2408.15664*.

[13] Nikita Kitaev, Åukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. In *International Conference on Learning Representations* (ICLR 2020).

[14] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, et al. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations* (ICLR 2020).

[15] Xuechen Li, Tianyi Zhang, Yann Dubois, et al. 2023. AlpacaEval: An Automatic Evaluator of Instruction-following Models. https://github.com/tatsu-lab/alpaca_eval

[16] Yang Liu, Dan Iter, Yichong Xu, et al. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2023).

[17] DeepSeek-AI. 2024. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv preprint arXiv:2405.04434*.

[18] Zechun Liu, Changsheng Zhao, Forrest Iandola, et al. 2024. MobileLLM: Optimizing Sub-billion Parameter Language Models for On-Device Use Cases.

Table 8: GPT-4 evaluation scores (1-10 scale) for generated stories. Mean ± std over 100 samples from 12L-1024d models. Inter-rater consistency measured using split-half correlation ($r = 0.87$).

Model	Grammar (↑)	Creativity (↑)	Coherence (↑)	Overall (↑)
MHA	7.1 ± 0.8	5.9 ± 1.2	5.6 ± 1.1	6.2 ± 0.9
MLA-RoPE	7.8 ± 0.7	7.2 ± 1.0	7.3 ± 0.9	7.4 ± 0.8
MoE-MHA	7.5 ± 0.7	6.8 ± 1.0	6.9 ± 0.9	7.1 ± 0.8
MoE-MLA-RoPE	**8.2 ± 0.6**	**7.9 ± 0.8**	**8.1 ± 0.7**	**8.1 ± 0.7**

In *International Conference on Machine Learning* (ICML 2024).

[19] Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning Sparse Neural Networks through L_0 Regularization. In *International Conference on Learning Representations* (ICLR 2018).

[20] Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? In *Advances in Neural Information Processing Systems* 32 (NeurIPS 2019).

[21] Asit Mishra, Jorge Albericio Latorre, Jeff Pool, et al. 2021. Accelerating Sparse Deep Neural Networks. *arXiv preprint arXiv:2104.08378*.

[22] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

[23] Timo Schick and Hinrich SchÃ¼tze. 2020. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL 2021).

[24] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, et al. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations* (ICLR 2017).

[25] Jianlin Su, Murtadha Ahmed, Yu Lu, et al. 2024. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing* 568:127063.

[26] Elena Voita, David Talbot, Fedor Moiseev, et al. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (ACL 2019).

[27] Sinong Wang, Belinda Z. Li, Madian Khabsa, et al. 2020. Linformer: Self-Attention with Linear Complexity. *arXiv preprint arXiv:2006.04768*.

[28] Fali Wang, Zhiwei Zhang, Xianren Zhang, et al. 2024. A Comprehensive Survey of Small Language Models in the Era of Large Language Models. *arXiv preprint arXiv:2411.03350*.

[29] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems* 36 (NeurIPS 2023).

[30] Deyao Zhu, Jun Chen, Xiaoqian Shen, et al. 2023. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. *arXiv preprint arXiv:2304.10592*.

[31] Sushant Mehta, Raj Dandekar, Rajat Dandekar, et al. 2023. Latent Multi-Head Attention for Small Language Models. *arXiv preprint arXiv:2506.09342*.

Are BabyLMs Deaf to Gricean Maxims? A Pragmatic Evaluation of Sample-efficient Language Models

Raha Askari[1,2], **Sina Zarrieß**[2], **Özge Alacam**[2], **Judith Sieker**[2]
[1]Department of Humanities, University of Turin
[2]Computational Linguistics, Department of Linguistics, Bielefeld University

Abstract

Implicit meanings are integral to human communication, making it essential for language models to be capable of identifying and interpreting them. Grice (1975) proposed a set of conversational maxims that guide cooperative dialogue, noting that speakers may deliberately violate these principles to express meanings beyond literal words, and that listeners, in turn, recognize such violations to draw pragmatic inferences. Building on Surian et al. (1996)'s study of children's sensitivity to violations of Gricean maxims, we introduce a novel benchmark to test whether language models pretrained on <10M and <100M tokens can distinguish maxim-adhering from maxim-violating utterances. We compare these BabyLMs across five maxims and situate their performance relative to children and a Large Language Model (LLM) pretrained on 3T tokens. We find that overall, models trained on <100M tokens outperform those trained on <10M, yet fall short of child-level and LLM competence. Our results suggest that modest data increases improve some aspects of pragmatic behavior, leading to finer-grained differentiation between pragmatic dimensions.

Our benchmark extends the BabyLM evaluation suite to pragmatic aspects of language and is publicly available.[1]

1 Introduction

Consider the following exchange: Sarah asks her friend *"What did you eat for lunch?"*, upon which her friend might reply *"I had something edible"* or *"I had chicken soup with an extra small silver spoon"*. While both responses are true and perfectly grammatical, the first one fails to provide the amount of information Sarah's question calls for, and the second one contains excessive, unasked details. Most listeners would expect an answer that is specific but not unnecessarily detailed, such as *"I had chicken soup"*.

In everyday conversation, such under- or over-informative replies stand out as odd because they do not provide the adequate amount of detail the question asks for. The philosopher Grice (1975) explained such phenomena through his *Cooperative principle*, which holds that speakers are generally aware of what is conversationally suitable or unsuitable. He proposed a set of conversational maxims, one of which, the *maxim of Quantity*, requires the speaker to be as informative as necessary.

The ability to notice and interpret such deviations from conversational norms is a key aspect of pragmatic competence, and essential for successful communication. In the evaluation of Large Language Models (LLMs), however, while models are now routinely tested on a wide range of syntactic tasks (e.g., Marvin and Linzen (2018); Hu et al. (2020); Finlayson et al. (2021); Lampinen (2024); Kryvosheieva and Levy (2025)), far fewer studies target their ability to reason pragmatically (Ettinger, 2020; Fried et al., 2023; Ma et al., 2025a; Sieker et al., 2025; Lachenmaier et al., 2025). This gap is especially pronounced for resource-limited models such as those developed for the BabyLM Challenge (Warstadt et al., 2023; Choshen et al., 2024). One possible reason is that, unlike syntax, pragmatics does not easily lend itself to large-scale minimal-pair test creation. Controlled operationalizations of pragmatic phenomena, such as those in Sieker et al. (2023) and Sieker and Zarrieß (2023), remain rare and resource-intensive, highlighting the challenge of designing systematic evaluation materials for this domain.

In psycholinguistics, however, several diagnostic tasks for pragmatic understanding already exist (e.g., Doran et al. (2012); Degen and Tanenhaus (2014); Romoli and Schwarz (2015); Tieu et al. (2015)). One such task is the Conversational Violations Test (CVT), designed to investigate children's

[1] https://huggingface.co/datasets/rahaaskari/gricean_baby

Proceedings of the First BabyLM Workshop, pages 52–65

pragmatic abilities based on Gricean maxims, introduced in Surian et al. (1996)'s study *"Are Children with Autism Deaf to Gricean Maxims?"*. In the CVT, children are presented with short dialogues where one answer follows one of Gricean conversational maxims and another violates it. Children are asked to identify the maxim-violating response. This controlled forced-choice format makes the CVT particularly attractive for LM evaluation: the correct choice depends on recognizing conversational norms rather than relying solely on factual knowledge or grammar, and the task fits well to established evaluation methods that compare model-assigned probabilities for a predefined set of candidate responses, as in grammatical acceptability (Warstadt et al., 2020), abductive commonsense reasoning (Zhao et al., 2023), or semantic relations of compound nouns (Rambelli et al., 2024).

In this paper, we adapt the CVT into a benchmark for evaluating BabyLMs' sensitivity to Gricean maxims. Starting from the original 25 conversational items from CVT, we augment the dataset automatically to over 2,250 items and refine them through human annotation. We evaluate a range of BabyLM baseline models (4 trained on <10M and 4 trained on <100M words), compare their performance to that of children from Surian et al. (1996), and situate their results alongside an LLM trained on more extensive data. In total, our experiment produces 20,250 data points (across 8 BabyLMs and 1 LLM). Among the evaluated models, BabyLMs trained on <100M tokens outperformed those trained on <10M, yet both groups fell short of achieving child-level pragmatic accuracy despite their developmental motivation. On average, BabyLMs performed best when judging truthfulness, but struggled most with assessing the appropriate level of informativeness. The LLM generally surpassed the BabyLMs and, in some cases, even outperformed children, but still failed to match children's overall competence. Thus, despite vastly larger training data, notable gaps persist between model and child performance across several maxims.

The contributions of this study are threefold: (1) a novel, linguistically controlled benchmark for testing pragmatic competence in data-limited LMs, grounded in established psycholinguistic methodology; (2) an empirical analysis of BabyLMs' performance across different Gricean maxims; and (3) a comparison of LM and child performance that situates model behavior within a developmental trajectory.

2 Background

Effective communication relies on more than just producing grammatical sentences. Much of what speakers communicate is conveyed implicitly, relying on the listener to infer meanings that go beyond the literal words. To do so successfully, speakers must choose utterances that are appropriate to the conversational context, and listeners must interpret them in light of shared assumptions, intentions and social norms. Even a perfectly well-formed sentence can be unhelpful, misleading or socially awkward if it ignores these unspoken rules. The study of how meaning is shaped by such contextual factors is the domain of pragmatics, which rests on the central idea that conversation is a cooperative activity: participants work together to exchange information efficiently and meaningfully. Grice (1975) formalized this intuition in his *Cooperative Principle*, according to which interlocutors are generally aware of what is conversationally suitable or unsuitable at each stage of a dialogue. He categorized this principle into four maxims, and additionally discussed Politeness as what he termed an "off-the-list" maxim:

- Quantity (*Make your contribution as informative as required and do not make your contribution more informative than is required*)
- Quality (*Do not say what you believe to be false and do not say that for which you lack adequate evidence*)
- Relation (*Be relevant*)
- Manner (*Be perspicuous, i.e., avoid obscurity, avoid ambiguity, be brief and be orderly*).
- Politeness (*Be polite*)

While these maxims are typically adhered to, speakers may sometimes blatantly violate them by saying one thing but implying another, producing what is known as an *implicature*. For example, when two colleagues are talking during a lunch break, one might ask *"Did you talk to the boss about the promotion?"*, and the other might reply, *"I really like this food."* This response violates the maxim of Relation and prompts the listener to search for the intended meaning, assuming the other person remains cooperative and aware of the maxims. In this case, for example, the interlocutor

is likely to infer that their colleague wishes to avoid the topic and has not spoken to the boss. Over the past decades, the Gricean maxims have become a cornerstone of pragmatic theory, shaping how researchers analyze and explain the ways people interpret and produce language in context.

Developmental Studies. Several developmental psycholinguistic studies have examined the age at which the sensitivity to such conversational violations emerges in humans (to name a few; Ackerman, 1981; Conti and Camras, 1984; Surian et al., 1996; Surian et al., 2010; Okanda et al., 2015 and Panzeri and Foppolo, 2021). In this direction, Surian et al. (1996)'s study introduced the Conversational Violations Test (CVT) to compare the pragmatic abilities of children with autism and specific language impairments to those of neurotypical children by incorporating Grice's framework. The maxims addressed in their study were Quantity (divided into two maxims; I: *Be informative* and II: *Avoid redundant information*), Quality, Relation and Politeness. The CVT is a set of 25 short conversational items and contains 5 conversations for each maxim. In their experiment, 8 neurotypical children (mean age 6-7) were presented with tape-recorded conversations featuring three puppets. In each scenario, one puppet would ask a question, and the other two would respond, only that one of them would provide an answer that violated a conversational maxim. The children were then asked to identify the puppet that gave the *silly or funny* answer, i.e., the one that violated the maxim. See Table 1 for examples for each maxim.

BabyLMs. The BabyLM Challenge aims to model human language development in order to optimize language model pretraining under strict data limitations. Tracks for submissions of text-only models include the Strict-small track (trained on <10M tokens) and the Strict track (trained on <100M tokens). As a starting point for evaluation, the organizers release a group of baseline models accompanied with pretraining corpora, along with an evaluation pipeline including benchmarks such as BLiMP (Warstadt et al., 2020) or GLUE (Wang et al., 2019)[2]. While these benchmarks provide broad coverage of linguistic competence, they do not directly and comprehensively assess pragmatic abilities. Given the developmental motivation behind BabyLMs, we argue that it is equally important to examine whether such models exhibit the pragmatic reasoning observed in humans.

[2]Find a complete overview of the BabyLM evaluation pipeline at https://github.com/babylm/evaluation-pipeline-2025.

Maxim	Example
Quantity I *Be informative*	**Q:** *How do you prefer your tea?* **Follower:** *With milk.* **Violator:** *In a cup.*
Quantity II *Avoid redundant information*	**Q:** *Who is your best friend?* **Follower:** *My best friend is John. He goes to my school.* **Violator:** *My best friend is Peter. He wears clothes.*
Quality *Be truthful*	**Q:** *Where do you live?* **Follower:** *I live in London.* **Violator:** *I live on the moon.*
Relation *Be relevant*	**Q:** *What games do you know?* **Follower:** *I know how to play football.* **Violator:** *I know your name.*
Politeness *Be polite*	**Q:** *Do you like my dress?* **Follower:** *It's pretty.* **Violator:** *I hate it.*

Table 1: Example items for different conversational maxims from Surian et al. (1996)'s CVT. The *Follower* adheres to the maxim, while the *Violator* does not.

Pragmatic Evaluation in LMs. As pragmatic knowledge is essential for successful communication, recent studies have explored whether LLMs exhibit pragmatic reasoning. Some studies report that LLMs can perform competitively with humans on specific tasks such as metaphor comprehension (Hu et al., 2023; Sanchez-Bayona and Agerri, 2025), but many find that they still struggle with a wide range of phenomena, including sarcasm and jokes (Hu et al., 2023; Jentzsch and Kersting, 2023), theory of mind (Shapira et al., 2023; Trott et al., 2023; Gandhi et al., 2024), implicit causality (Sieker et al., 2023; Kankowski et al., 2025), context-dependent reference resolution (Junker et al., 2025; Ma et al., 2025b), or inferences like presuppositions (Kabbara and Cheung, 2022; Sieker and Zarrieß, 2023; Tsvilodub et al., 2024; Sieker et al., 2025; Lachenmaier et al., 2025).

When it comes to the Gricean maxims, Hu et al. (2023), for example, evaluated LLMs' ability to understand intended meanings by prompting models with short English stories and asking what a char-

acter wanted to convey by flouting a maxim, given a set of possible answers. They found that the models would generally assign higher probabilities to literal meanings over the speaker's intended meaning. Similarly, LLMs demonstrated bias towards literal meanings during a pragmatic evaluation for Korean language by Park et al. (2024). In their experiment, models performed poorly on the maxim of Relation and well on the maxim of Quality when selecting the pragmatic meanings from given options, but showed reversed patterns for open-ended questions about the speaker's intent. Yue et al. (2024), on the other hand, evaluated models' ability to infer implicated meanings in multi-turn Chinese dialogues and found no significant variation in model performances across maxims. Moreover, most models failed to generate correct interpretations for implicatures despite being able to identify them in a multiple-choice setting. Other examples of pragmatic evaluations of LLMs by incorporating implicatures include (but are not limited to) Zheng et al. (2021), who presented the GRICE dataset for assessing the pragmatic reasoning of LLMs while taking into account other aspects of modern dialogue modeling like coreference; Cho and Kim (2024), who compared cosine similarities of literal meanings of scalar implicatures with their pragmatic meanings; and Kurch et al. (2024), who tested whether LLMs can derive atypicality inferences that are triggered through information redundancy.

Building on this line of work, we extend prior studies on LM pragmatic competence and Gricean maxims by introducing a child-directed, maxim-balanced benchmark that enables direct comparison between model and child performance. Inspired by Surian et al. (1996)'s CVT, we compile a dataset of 2,250 conversational items in a controlled forced-choice format. Our benchmark is particularly well-suited to the BabyLM Challenge because its simple, child-appropriate language and controlled design offer a fine-grained, diagnostic test of pragmatic abilities, while minimizing reliance on large-scale training data or extensive world knowledge.

3 Approach

In order for pragmatic interpretations (those that go beyond literal ones) to arise, a listener must know the rules of conversation, recognize when they are violated, and discern when an utterance may be literally unfitting (uncooperative) yet pragmatically acceptable (cooperative). Building on the framework of Gricean maxims, Surian et al. (1996) examined the pragmatic competence of children by testing whether they could identify an uncooperative (i.e., maxim-violating) answer among a pair of responses to a given question. We adopt this same forced-choice paradigm to evaluate the pragmatic sensitivity of language models.

Data. We base our evaluation on the CVT and extend this resource into a large-scale benchmark by generating additional CVT-style items with GPT-4 (OpenAI, 2024) and manually curating the outputs to maintain child-level vocabulary[3], grammaticality, naturalness and adherence to the targeted maxim. The final dataset contains 2,250 dialogues, balanced across five maxims: Quantity I, Quantity II, Quality, Relation, and Politeness. Full details of the augmentation process and quality control criteria are provided in Appendix A.1. Also, see Appendix A.2 for examples of experimental items of our dataset.

Models. We use the following baseline BabyLMs pretrained on BabyLM corpora that were released in two tracks (Strict for models trained on at most 100M tokens and Strict-small for models trained on at most 10M tokens)[4]: two auto-regressive LMs, namely GPT-2 (Radford et al., 2019) and Baby Llama (Timiryasov and Tastet, 2023) and two masked LMs, namely LTG-BERT (Samuel et al., 2023) and Roberta (Liu et al., 2019). Finally, to assess the effect of more training data on the pragmatic performance of language models and to enable a comparison with an LLM, we evaluate the decoder-only OLMo-1B (Groeneveld et al., 2024) as a representative of fully open LLMs that has been trained on 3T tokens.

Evaluation. Using our curated dataset, we evaluate pragmatic sensitivity of language models in an unsupervised setting. Specifically, we measure a model's sentence acceptability for the two candidate answers to a question: one that follows a Gricean maxim (follower) and one that violates it (violator). For incremental models, we compute the conditional log-probability of the answer given the question, while in the case of

[3]This is derived from the fact that the pretraining data for baseline BabyLMs mostly consists of input received by children.

[4]Baseline models for previous years and this year's submission are available at https://huggingface.co/babylm and https://huggingface.co/BabyLM-community.

masked language models, we use the improved pseudo-log-likelihood proposed by Kauf and Ivanova (2023). In both cases, the probability of the answer is calculated as the sum of the log-probabilities of its tokens, normalized by its length. For each item, we assess whether the model assigns a higher probability to the maxim-following answer:

$$\mathbb{1}[P(\text{Answer}_{\text{Follower}} \mid \text{Question}) > P(\text{Answer}_{\text{Violator}} \mid \text{Question})]$$

Model accuracy for each maxim is defined as the proportion of items for which the model assigns a higher probability to the maxim-follower response. We obtain models' scores through Minicons (Misra, 2022)[5], which is is an open-source library for extracting sentence acceptability measures in language models.

4 Results

In Table 2, we report accuracy per maxim for the BabyLM baselines in the Strict-small and Strict tracks. Furthermore, we present the results from OLMo-1B. Finally, as a reference point, we include the results of children who were tested on the CVT by Surian et al. (1996). In the following, we break down the results by conversational maxim, model architecture and model size.

Results by Gricean Maxim. As shown in Table 2, model performance varies considerably across different maxims in both the Strict-small and Strict tracks. The maxims Quantity I (*Be informative*) and Quantity II (*Avoid redundant information*) are consistently the most challenging, with average accuracies for all BabyLMs peaking at only 0.59. In contrast, Quality (*Be truthful*) emerges as the easiest category for most BabyLMs with the average accuracy as high as 0.74. Relation (*Be relevant*) and Politeness (*Be polite*) generally fall in between these extremes (although exceptions apply), with the average accuracies above chance but below the best Quality results.

This pattern suggests that factuality is easier for BabyLMs to capture from limited data, likely because it can be learned from explicit statements and lexical associations in the training data, whereas judgments of informativeness and redundancy require more context-sensitive reasoning. The representative examples from the original CVT in Table 1 illustrate this: in Quantity I, while *"with milk"* falls in the range of pragmatically accepted answers, *"in a cup"* might contain tokens (or token combinations) that are more frequent in the training data. Across the other maxims, such frequent continuations may similarly override pragmatic appropriateness in model predictions.

To quantify how consistently models agree on these difficulty patterns, we ranked maxims per model and computed Kendall's W. Agreement was high in the Strict-small track ($W = 0.80$) and moderate in the Strict track ($W = 0.68$), indicating that models trained on less data tend to exhibit more similar difficulty patterns. In both tracks, the maxims Quantity I and Quantity II were generally ranked as the hardest maxims. Full rankings and statistics are reported in Table 3.

Inter-maxim Correlations. To examine whether performance on different maxims co-varies across models, we computed Pearson correlations between per-model accuracies across maxims for each track (Figure 1). In the Strict-small track, correlations between maxims tend to be extreme. Quantity I and Quantity II show an almost perfect correlation ($r = 0.953$). The same track also reveals a striking near-perfect correlation between Relation and Politeness ($r = 0.999$), suggesting that topicality and politeness violations may be treated in similar ways. In contrast, Quality stands out in the Strict-small track as largely decoupled from the other maxims (near-zero or negative correlations), which may indicate that detecting literal implausibility (e.g., *"on the moon"*, Table 1) behaves independently of other pragmatic abilities under data constraints. In the Strict track, correlations are overall weaker and more varied, with some negative associations appearing for pairs that were strongly positive in the Strict-small track (e.g., Politeness vs. Quantity I, ($r = -0.155$), possibly reflecting a partial decoupling of politeness from topicality and informativeness once models have more data. Notably, the maxim of Quality strongly correlates with Quantity I in the Strict track ($r = 0.931$), despite being among the easiest maxims in difficulty rankings (Table 3), showing that correlation patterns capture co-variation rather than absolute difficulty. Overall, the shift from rather extreme relations in the Strict-small track to more varied and generally weaker associations in the Strict track complements the prior results, suggesting that with more training data, models begin to differentiate more between pragmatic

[5]Available at https://github.com/kanishkamisra/minicons.

	Quantity I	Quantity II	Quality	Relation	Politeness	Overall
Strict-small BabyLMs						
GPT-2	**0.59**	**0.59**	0.61	**0.66**	**0.80**	**0.65**
Baby Llama	0.56	0.49	0.74	0.63	0.68	0.62
LTG BERT	0.45	0.45	**0.76**	0.58	0.46	0.54
Roberta	0.33	0.36	0.64	0.60	0.57	0.50
Average	0.48	0.47	0.69*	0.61	0.63	
Strict BabyLMs						
GPT-2	0.60	0.68	0.76	0.72	**0.76**	**0.70**
Baby Llama	0.55	**0.64**	0.75	0.67	0.70	0.66
LTG BERT	**0.64**	0.57	**0.79**	**0.74**	0.52	0.65
Roberta	0.51	0.47	0.67	0.68	0.59	0.58
Average	0.58	0.59	0.74*	0.70	0.64	
LLM						
OLMo-1B	0.76	0.83	0.83	0.84	0.67	0.79
Children						
	0.58	0.78	1.0	1.0	0.93	0.86

Table 2: Accuracy scores across the Gricean maxims. *Strict-small* models are pretrained on <10M tokens and *Strict* models on <100M. OLMo-1B is pretrained on 3T tokens. All models are evaluated on 2,250 items, while child accuracy scores are from 8 neurotypical children from Surian et al. (1996). For the Strict-small and Strict groups, the highest score of each maxim is bolded. The highest average across all maxims is marked with (*).

Model	Quant. I	Quant. II	Qual.	Rel.	Polite
Strict-small					
GPT-2	🟠	🔴	🟡	🟢	🟢
Baby Llama	🟠	🔴	🟢	🟡	🟢
LTG BERT	🟠	🔴	🟢	🟢	🟡
RoBERTa	🔴	🟠	🟢	🟢	🟡
Strict					
GPT-2	🔴	🟠	🟢	🟡	🟢
Baby Llama	🔴	🟠	🟢	🟡	🟢
LTG BERT	🟡	🟠	🟢	🟢	🔴
RoBERTa	🟠	🔴	🟡	🟢	🟢

Table 3: Maxim difficulty rankings for each model (🟢 = easiest, 🔴 = hardest). Kendall's W: Strict-small = 0.80, $\chi^2(4) = 12.80$, $p = 0.012$; Strict = 0.68, $\chi^2(4) = 10.80$, $p = 0.029$.

dimensions.

Effects of Architecture and Model size. Referring to Table 2, it is further notable that across both tracks, GPT-2 consistently outperforms LTG-BERT and RoBERTa models, with BabyLLaMA typically ranking second. RoBERTa BabyLM performs worst overall, particularly on the Quantity maxims. Increasing the training data from the Strict-small (<10M tokens) to the Strict (<100M tokens) track generally improves performance, especially for Quantity I and Quantity II, where average scores increase by 0.09–0.12. Gains for Quality and Relation are more modest, while Politeness scores remain similar across tracks. Interestingly, GPT-2's Politeness score decreases in the Strict track, suggesting that greater exposure to varied language might introduce alternative patterns that increase uncertainty in politeness judgments.

The large-scale OLMo 1B model substantially outperforms all BabyLM baselines across the maxims, with the exception of GPT-2 Strict-small, which scored 0.80 in Politeness. OLMo 1B scored substantially higher on the maxim of Relation compared to BabyLMs, indicating that sensitivity to topic relevance tends to emerge with increased training data.

The differences of scores may also reflect how auto-regressive and masked LMs handle conversational flow: in our setup, probability assignment to an entire answer benefits from modeling sequences as coherent continuations rather than token-masked completions. The performance gap between Strict-small and Strict models also indicates that increased training data helps, but does not eliminate the persistent difficulties with Quantity-related judgments.

Comparison to Child Performance. In Surian et al. (1996), neurotypical children showed an overall high accuracy (0.86), likely reflecting the development of Theory of Mind (Baron-Cohen et al., 1985), the impairment of which damages recognizing speaker's intended meaning. BabyLMs share some similarities with children: like them, they perform best on the maxim of Quality and worst on the Quantity maxims (Table 2). However, they do not demonstrate the the same high performance

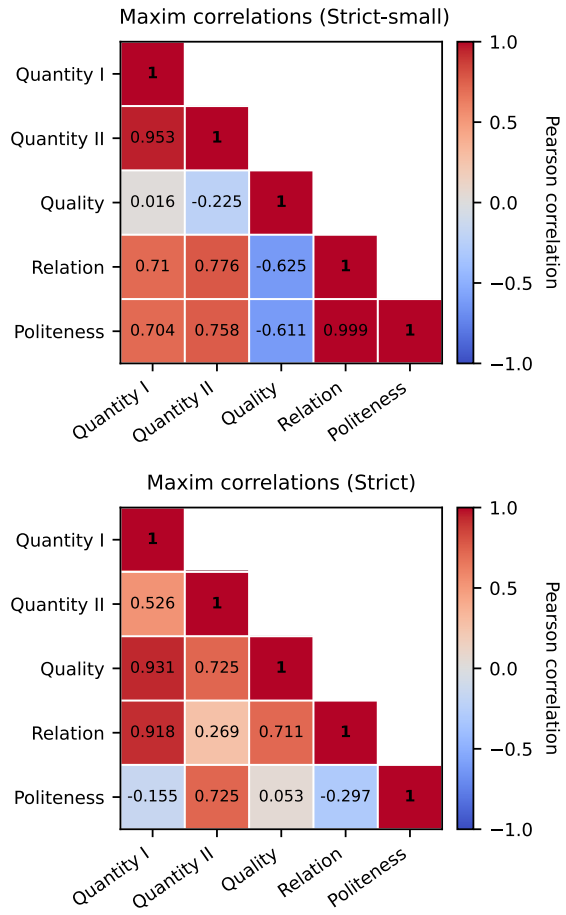


Figure 1: Pearson correlations between per-model accuracies across maxims. Top: Strict-small track. Bottom: Strict track.

as children in maxims of Relation and Politeness. Furthermore, children's overall high accuracy indicates that, by school age, they are already highly sensitive to conversational norms. In contrast, BabyLMs reach only 0.50–0.70 of overall accuracy, underscoring a substantial gap in pragmatic competence between small-scale LMs and six-to-seven-year-old human speakers. The LLM shows a different pattern: it surpasses children on the Quantity maxims, indicating stronger performance on informativeness-related judgments, but still falls short on other maxims, especially Politeness, suggesting that socially grounded pragmatic norms do not emerge automatically from large-scale pretraining. Overall, both small-scale and large-scale models reveal persistent limitations in capturing the full range of conversational norms.

Summary and Discussion. Across both data tracks, the greatest deficits in BabyLM performance concerned judgments of informativeness required for an appropriate response. The maxim of Quality was the easiest, with Relation and Politeness in between (Tables 2 and 3). Correlation analyses further revealed that, under severe data constraints, most maxims were strongly associated with at least one other (e.g., Quantity I and II; Relation and Politeness) (Figure 1). These associations tended to weaken with more data, indicating a shift toward more differentiated treatment of pragmatic categories. Among architectures, GPT-2 performed best overall, RoBERTa worst, and scaling from <10M to <100M tokens yielded the largest gains on Quantity, though sometimes at the expense of Politeness (Table 2), suggesting that autoregressive modeling and modest scaling benefit informativeness but may reduce social-pragmatic sensitivity. Compared to children, (0.86 overall, Table 2), BabyLMs mirrored the relative ordering of difficulty but scored substantially lower (0.50–0.70). The LLM (OLMo-1B) outperformed BabyLMs' overall performance in all maxims and exceeded child performance on Quantity, yet lagged considerably on the remaining maxims, showing that large-scale pretraining enhances information-structuring abilities but offers limited gains in other dimensions of pragmatic understanding.

Our results align with earlier findings that pragmatic competence in language models scales with model size and training data but may remain below human levels. For instance, Hu et al. (2023) reported that GPT-2 with 117M parameters did not perform above chance when interpreting maxim-flouting utterances, in line with our observation that BabyLMs trained on <100M tokens perform rather poorly across maxims. At the other end of the scale, Yue et al. (2024) found that LLaMA 2 models with 13B parameters performed above chance across several Gricean maxims but still achieved only about half the human score, while GPT-4 matched human performance. In this context, our results with OLMo-1B suggest that large-scale pretraining can surpass child performance on informativeness but still leaves substantial gaps on more socially grounded maxims such as Politeness.

Overall, these findings indicate that scaling data and parameters improves some aspects of pragmatic reasoning in language models, while their absolute performance remains far from child-like. This underscores the importance of dedicated evaluation benchmarks targeting pragmatic abilities, ensuring that the developmental goals of the BabyLM challenge address this crucial aspect of language

use.

5 Conclusion

This paper introduced a novel large-scale benchmark for evaluating the pragmatic competence of language models, grounded in the Gricean maxims and adapted from a psycholinguistic test suite. Using 2,250 conversational items, we assessed BabyLM baseline models trained on constrained data alongside a large-scale 3T-token model and projected their performance on that of children. Our results indicate that increasing training data from <10M to <100M tokens leads to performance gains, yet BabyLMs remain below child-level competence. They demonstrated the lowest performance in assessing the appropriate amount of information for conversationally acceptable responses and did not exhibit a preference for answers that were more polite or contextually relevant. Furthermore, correlation analyses revealed that under data-limited conditions, models tend to conflate certain pragmatic competences, but these associations weaken with more data, suggesting that additional exposure allows models to more clearly differentiate between distinct pragmatic dimensions.

Our benchmark provides a linguistically grounded, scalable evaluation resource that enables systematic and comparable measurement of pragmatic behavior across models of different sizes and training regimes. By extending the BabyLM evaluation suite with a dedicated pragmatic benchmark, this work provides a tool for systematically tracking progress on this essential aspect of human-like language understanding.

Limitations and Future Directions. In this section, we state the limitations of our study and possible directions they offer for future work.

First, unlike other datasets (Zheng et al., 2021; Hu et al., 2023; Park et al., 2024), the conversational items in our dataset do not include detailed scenarios that are embedded before prompting models with dialogues. In certain contexts, the maxim-violator responses in our dataset could be in fact appropriate; for instance, the answer *"My best friend is Peter. He wears clothes."* (Table 1) would not be redundant in a scenario where others are unclothed. However, even within a minimal-context setting like the one implemented in this study, non-linguist participants have been shown to consistently favor responses that adhere to conversational maxims. For example, Okanda et al. (2015) applied a revised Japanese version of the CVT and found that adults were able to identify non-cooperative answers and articulate the reasoning behind their judgments. Nevertheless, future work could expand our conversational items by introducing explicit context that would render maxim-violator responses cooperative (as in the above-mentioned example) to examine whether such framing would change model preferences.

Second, we acknowledge that model probability assignments may be influenced by the distributional properties of tokens independent of context, which can make evaluations based on sequence scores prone to bias. Future work could address this by expanding the dataset to include a wider range of lexical variations.

Third, we selected the 1B-parameter OLMo model due to computational constraints. Although our primary focus was on BabyLMs rather than large models, evaluating systems of varying sizes offers valuable comparative insights. Furthermore, as our results suggest that model architecture affects pragmatic performance, future work could test whether these patterns hold for other architectures, such as instruction-tuned or multimodal (Baby) models.

Finally, our dataset is currently limited to English; therefore, extending this evaluation to multilingual settings would allow for more robust conclusions and enable meaningful cross-linguistic comparisons of pragmatic competence.

Ethical Statement

Our work uses publicly available data from a psycholinguistic study on children; we do not conduct any new experiments involving human subjects. No personal information from the original study is shared, except for the mean age of participants as reported by the authors. We do not train any new models; instead, our evaluation dataset was automatically generated and subsequently reviewed and refined by two of the authors. The dataset consists of short, child-level conversational exchanges, and we believe that none of the items raise ethical concerns or reinforce biases toward sensitive groups. Our evaluation focuses exclusively on the pragmatic competence of language models. We do not address other potential harms or limitations of these systems such as discrimination, toxicity and misinformation Weidinger et al. (2021) which remain important areas for continued investigation and responsible deployment. We share our evalu-

ation dataset, code and model results publicly to facilitate future use and promote transparency.

Acknowledgements

We warmly thank the anonymous reviewers for their insightful comments. We acknowledge support from several projects and funding institutions: 1) “SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), an initiative of the Ministry of Culture and Science of the State of Northrhine Westphalia; 2) Erasmus+, a European Union funding that facilitates periods of study or training abroad, 3) Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – CRC-1646, project number 512393437, project B02.

References

Brian P Ackerman. 1981. When is a question not answered? the understanding of young children of utterances violating or conforming to the rules of conversational sequencing. *Journal of Experimental Child Psychology*, 31(3):487–507.

Simon Baron-Cohen, Alan M. Leslie, and Uta Frith. 1985. Does the autistic child have a “theory of mind” ? *Cognition*, 21(1):37–46.

Ye-eun Cho and Seong mook Kim. 2024. Pragmatic inference of scalar implicature by LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 10–20, Bangkok, Thailand. Association for Computational Linguistics.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [call for papers] the 2nd babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2404.06214.

Daniel J Conti and Linda A Camras. 1984. Children’s understanding of conversational principles. *Journal of Experimental Child Psychology*, 38(3):456–463.

Judith Degen and Michael Tanenhaus. 2014. Processing scalar implicature: A constraint-based approach. *Cognitive Science*.

Ryan Doran, Gregory Ward, Meredith Larson, Yaron McNabb, and Rachel E Baker. 2012. A novel experimental paradigm for distinguishing between what is said and what is implicated. *Language*, 88(1):124–154.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics.

Daniel Fried, Nicholas Tomlin, Jennifer Hu, Roma Patel, and Aida Nematzadeh. 2023. Pragmatics in language grounding: Phenomena, tasks, and modeling approaches. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12619–12640, Singapore. Association for Computational Linguistics.

Kanishk Gandhi, J.-Philipp Fränken, Tobias Gerstenberg, and Noah D. Goodman. 2024. Understanding social reasoning in language models with language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS ’23, Red Hook, NY, USA. Curran Associates Inc.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Sophie Jentzsch and Kristian Kersting. 2023. ChatGPT is fun, but it is not funny! humor is still challenging large language models. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 325–340, Toronto, Canada. Association for Computational Linguistics.

Simeon Junker, Manar Ali, Larissa Koch, Sina Zarrieß, and Hendrik Buschmeier. 2025. Are multimodal large language models pragmatically competent listeners in simple reference resolution tasks? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24101–24109, Vienna, Austria. Association for Computational Linguistics.

Jad Kabbara and Jackie Chi Kit Cheung. 2022. Investigating the performance of Transformer-Based NLI models on presuppositional inferences. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 779–785, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Florian Kankowski, Torgrim Solstad, Sina Zarrie§, and Oliver Bott. 2025. Instruction tuning modulates discourse biases in language models.

Carina Kauf and Anna Ivanova. 2023. A better way to do masked language model scoring. *Preprint*, arXiv:2305.10588.

Daria Kryvosheieva and Roger Levy. 2025. Controlled evaluation of syntactic knowledge in multilingual language models. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 402–413, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Charlotte Kurch, Margarita Ryzhova, and Vera Demberg. 2024. Large language models fail to derive atypicality inferences in a human-like manner. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 86–100, Bangkok, Thailand. Association for Computational Linguistics.

Clara Lachenmaier, Judith Sieker, and Sina Zarrieß. 2025. Can LLMs ground when they (don't) know: A study on direct and loaded political questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14956–14975, Vienna, Austria. Association for Computational Linguistics.

Andrew Lampinen. 2024. Can language models handle recursively nested grammatical structures? a case study on comparing models and humans. *Computational Linguistics*, 50(3):1441–1476.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Bolei Ma, Yuting Li, Wei Zhou, Ziwei Gong, Yang Janet Liu, Katja Jasinskaja, Annemarie Friedrich, Julia Hirschberg, Frauke Kreuter, and Barbara Plank. 2025a. Pragmatics in the era of large language models: A survey on datasets, evaluation, opportunities and challenges. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8679–8696, Vienna, Austria. Association for Computational Linguistics.

Ziqiao Ma, Jing Ding, Xuejun Zhang, Dezhi Luo, Jiahe Ding, Sihan Xu, Yuchen Huang, Run Peng, and Joyce Chai. 2025b. Vision-language models are not pragmatically competent in referring expression generation. *Preprint*, arXiv:2504.16060.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Kanishka Misra. 2022. minicons: Enabling flexible behavioral and representational analyses of transformer language models. *Preprint*, arXiv:2203.13112.

Mako Okanda, Kosuke Asada, Yusuke Moriguchi, and Shoji Itakura. 2015. Understanding violations of gricean maxims in preschoolers and adults. *Frontiers in psychology*, 6:901.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Francesca Panzeri and Francesca Foppolo. 2021. Children's and adults' sensitivity to gricean maxims and to the maximize presupposition principle. *Frontiers in Psychology*, 12:624628.

Dojun Park, Jiwoo Lee, Hyeyun Jeong, Seohyun Park, and Sungeun Lee. 2024. Pragmatic competence evaluation of large language models for the Korean language. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, pages 256–266, Tokyo, Japan. Tokyo University of Foreign Studies.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Giulia Rambelli, Emmanuele Chersoni, Claudia Collacciani, and Marianna Bolognesi. 2024. Can large language models interpret noun-noun compounds? a linguistically-motivated study on lexicalized and novel compounds. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11823–11835, Bangkok, Thailand. Association for Computational Linguistics.

Jacopo Romoli and Florian Schwarz. 2015. *An Experimental Comparison Between Presuppositions and Indirect Scalar Implicatures*, pages 215–240. Springer International Publishing, Cham.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Elisa Sanchez-Bayona and Rodrigo Agerri. 2025. Metaphor and large language models: When surface features matter more than deep understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17462–17477, Vienna, Austria. Association for Computational Linguistics.

Natalie Shapira, Guy Zwirn, and Yoav Goldberg. 2023. How well do large language models perform on faux pas tests? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10438–10451, Toronto, Canada. Association for Computational Linguistics.

Judith Sieker, Oliver Bott, Torgrim Solstad, and Sina Zarrieß. 2023. Beyond the bias: Unveiling the quality of implicit causality prompt continuations in language models. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 206–220, Prague, Czechia. Association for Computational Linguistics.

Judith Sieker, Clara Lachenmaier, and Sina Zarrie§. 2025. LLMs struggle to reject false presuppositions when misinformation stakes are high. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 47.

Judith Sieker and Sina Zarrieß. 2023. When your language model cannot Even do determiners right: Probing for anti-presuppositions and the maximize presupposition! principle. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 180–198, Singapore. Association for Computational Linguistics.

Luca Surian, Simon Baron-Cohen, and Heather van der Lely. 1996. Are children with autism deaf to gricean maxims? *Cognitive neuropsychiatry*, 1:55–72.

Luca Surian, Mariantonia Tedoldi, and Michael Siegal. 2010. Sensitivity to conversational maxims in deaf and hearing children. *Journal of Child Language*, 37(4):929–943.

Lyn Tieu, Jacopo Romoli, Peng Zhou, and Stephen Crain. 2015. Children's knowledge of free choice inferences and scalar implicatures. *Journal of Semantics*, 33(2):269–298.

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. *Preprint*, arXiv:2308.02019.

Sean Trott, Cameron Jones, Tyler Chang, James Michaelov, and Benjamin Bergen. 2023. Do large language models know what humans know? *Cognitive Science*, 47(7):e13309.

Polina Tsvilodub, Paul Marty, Sonia Ramotowska, Jacopo Romoli, and Michael Franke. 2024. Experimental pragmatics with machines: Testing llm predictions for the inferences of plain and embedded disjunctions. *Preprint*, arXiv:2405.05776.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, and 1 others. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Shisen Yue, Siyuan Song, Xinyuan Cheng, and Hai Hu. 2024. Do large language models understand conversational implicature- a case study with a Chinese sitcom. In *Proceedings of the 23rd Chinese National Conference on Computational Linguistics (Volume 1: Main Conference)*, pages 1270–1285, Taiyuan, China. Chinese Information Processing Society of China.

Wenting Zhao, Justin Chiu, Claire Cardie, and Alexander Rush. 2023. Abductive commonsense reasoning exploiting mutually exclusive explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14883–14896, Toronto, Canada. Association for Computational Linguistics.

Zilong Zheng, Shuwen Qiu, Lifeng Fan, Yixin Zhu, and Song-Chun Zhu. 2021. GRICE: A grammar-based dataset for recovering implicature and conversational rEasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2074–2085, Online. Association for Computational Linguistics.

A Appendix

A.1 Dataset Augmentation Details

In order to expand CVT for model evaluation, we employed the GPT-4 chat interface (OpenAI, 2024) to synthesize similar items. First, we provided GPT-4 with a brief description of the maxims and examples from the CVT, and asked it to generate 25 corresponding dialogue templates, each accompanied by two paraphrased versions of the question. Answers were unchanged in the paraphrased versions; this was due to some answers being too short to paraphrase, and we chose a unified method for all items. Next, each template was presented with its original CVT example, and GPT-4 was asked to produce four new conversations per template, using vocabulary appropriate for children. Finally, the paraphrased versions of both the original CVT items and the newly created conversations were generated based on their respective templates. This resulted in a dataset consisting of 25 CVT items and 350 GPT-generated ones. Table 4 shows a concrete example of this pipeline.

Two of the authors manually reviewed all 375 items and made adjustments based on the following criteria:

- The follower's answer does not follow the maxim.
- The violator's answer does not violate the maxim.
- The paraphrased versions do not correctly paraphrase the questions.
- The words exceed child-level vocabulary.
- The questions and/or answers are unnatural or ungrammatical.
- The answers are inadequate for model evaluation.

The last criterion reflects our effort to minimize superficial differences between the follower and violator answers wherever possible. Since our evaluation compares model probabilities for each answer pair, irrelevant lexical differences would distort the results. For example, in response to the question *Who is your best friend?*, if one answer was *My best friend is **John**. He goes to my school.* and the other was *My best friend is **Peter**. He wears clothes.*, not replacing the names with a single name would introduce noise unrelated to pragmatic reasoning. However, in some cases, such standardizations were not applicable due to the nature of the maxims being tested; for example, in the case of the maxim of Relation, altering the content is necessary for the violation.

Once approved, we diversified the dataset by assigning speaker names (Leslie, Joan, and Thomas) and rotating them such that each conversation had one name as the questioner and another as the responder, used consistently across both responses. The names were randomly selected as English common names from the firstname database at `https://github.com/KarlAmort/firstname-database`. This process yielded a final dataset of 2,250 conversational items, since six variations with names were generated from each of the 375 curated conversations.

Stage	Example
1. GPT-4 is prompted with a dialogue from CVT	Question: *What would you like to buy in this shoe-shop?* Follower: *A pair of trainers.* Violator: *A pair of shoes.*
2. GPT-4 generates a template from the dialogue	Question: *What would you like to buy in this* `[type of store]`? Follower (specific, informative): `[A concrete item you might buy in that specific store]` Violator (vague, unhelpful): `[A general category that lacks specific information]`
3. GPT-4 generates four new dialogues from the template of stage 2	1. Question: *What would you like to buy in this toy shop?* Follower: *A teddy bear.* Violator: *A toy.* 2. Question: *What would you like to buy in this bakery?* Follower: *A chocolate muffin.* Violator: *Something to eat.* 3. Question: *What would you like to buy in this bookshop?* Follower: *A Harry Potter book.* Violator: *A book.* 4. Question: *What would you like to buy in this clothes shop?* Follower: *A red jacket.* Violator: *Some clothes.*
4. GPT-4 generates 2 paraphrased questions of the template of stage 2	1. *Is there something you'd like to get from this* `[type of store]`? 2. *What are you looking for in this* [type of store]?
5. GPT-4 generates 2 paraphrased versions of the original CVT dialogue given in stage 1 and all other dialogues generated in stage 3	1. Question: *Is there something you'd like to get from this shoe-shop?* Follower: *A pair of trainers.* Violator: *A pair of shoes.* 2. Question: *What are you looking for in this shoe-shop?* Follower: *A pair of trainers.* Violator: *A pair of shoes.* And so on

Table 4: An example of the data augmentation pipeline for the maxim of Quantity I (*Be informative*). The Follower adheres to the maxim, while the Violator does not. For each dialogue in CVT, four more examples with child-level vocabulary were created. Later, two paraphrased versions for all dialogues (those from CVT and GPT-generated ones) were synthesized and added.

A.2 Dataset Examples

We depict a few examples from the dataset. The maxim-violator's answer is marked with (*).

Maxim of Quantity I (Be informative):

- Leslie: What did you eat for supper?
  Thomas: Tomato soup.
  Thomas: A dish.*

- Leslie: What did you see at the zoo?
  Joan: The lions.
  Joan: Some animals.*

- Leslie: What did you get for Christmas?
  Joan: A gift.*
  Joan: A toy train.

- Joan: How do you prefer your pancakes?
  Leslie: On a plate.*
  Leslie: With maple syrup.

Maxim of Quantity II (Avoid redundant information):

- Leslie: Who is your neighbor?
  Thomas: My neighbor is Mr. Tom. He has a dog.
  Thomas: My neighbor is Mr. Tom. He lives in a house.*

- Joan: What pet do you like?
  Leslie: I like puppies and kittens which are pets.*
  Leslie: I like puppies and kittens which are cute.

- Leslie: Where did you go last weekend?
  Joan: I went to grandma's house and baked cookies.
  Joan: I went to grandma's house and I didn't stay in my room.*

- Joan: Which is your favourite fruit?
  Thomas: Watermelon which is a fruit.*

Thomas: Watermelon which is juicy.

Maxim of Quality (Be truthful):

- Thomas: Is there any more popcorn?
  Leslie: Yes, there's a bowl in the kitchen.
  Leslie: Yes, it's raining popcorn outside.*
- Leslie: Where do you do your homework?
  Joan: I do them on a dragon's back.*
  Joan: I do them in my room.
- Joan: Do you have any pets?
  Leslie: Yes, I have a cat and a fish.
  Leslie: Yes, I have a thousand elephants.*
- Leslie: Why don't you come outside?
  Thomas: Because I'm helping mom bake.
  Thomas: Because I'm on a spaceship.*

Maxim of Relation (Be relevant):

- Joan: What did you do on the weekend?
  Thomas: I went to the zoo.
  Thomas: My socks are green.*
- Thomas: What do you like to play?
  Leslie: I like to play tag.
  Leslie: I like chocolate cake.*
- Joan: What is your favourite animal?
  Thomas: I like pencils best.*
  Thomas: I like pandas best.
- Thomas: What songs do you know?
  Joan: I know "Twinkle Twinkle Little Star.
  Joan: I know how to tie my shoes.*

Maxim of Politeness (Be polite):

- Thomas: Do you like my new haircut?
  Joan: It looks awful.*
  Joan: It looks nice.
- Thomas: Would you like to try some of my cake?
  Joan: No, thanks.
  Joan: No, it's disgusting.*
- Joan: May I use your calculator?
  Thomas: No, don't touch my stuff.*
  Thomas: No, sorry, I need it right now, but you can use it after.
- Leslie: Could you help me with my puzzle?
  Thomas: Do it by yourself.*
  Thomas: Sure, after I finish this one.

Model Merging to Maintain Language-Only Performance in Developmentally Plausible Multimodal Models

Ece Takmaz Lisa Bylinina Jakub Dotlačil
Utrecht University
{e.k.takmaz | e.g.bylinina | j.dotlacil}@uu.nl

Abstract

State-of-the-art vision-and-language models consist of many parameters and learn from enormous datasets, surpassing the amounts of linguistic data that children are exposed to as they acquire a language. This paper presents our approach to the multimodal track of the BabyLM challenge addressing this discrepancy. We develop language-only and multimodal models in low-resource settings using developmentally plausible datasets, with our multimodal models outperforming previous BabyLM baselines. One finding in the multimodal language model literature is that these models tend to underperform in *language-only* tasks. Therefore, we focus on maintaining language-only abilities in multimodal models. To this end, we experiment with *model merging*, where we fuse the parameters of multimodal models with those of language-only models using weighted linear interpolation. Our results corroborate the findings that multimodal models underperform in language-only benchmarks that focus on grammar, and model merging with text-only models can help alleviate this problem to some extent, while maintaining multimodal performance.

1 Introduction

Current state-of-the-art multimodal language models (MLMs) are composed of many layers containing billions of parameters and they require huge amounts of data to learn how to handle and bridge visual and textual modalities. On the other hand, children acquire language with the help of much smaller sets of linguistic input. The BabyLM challenge (Warstadt et al., 2023) focuses on this discrepancy and encourages the implementation and training of sample-efficient, developmentally plausible models in resource-limited contexts. Although utilizing small datasets and models could prove challenging to outperform current MLMs, such setups could allow for cognitive plausibility, also making the development and use of such models more accessible and efficient.

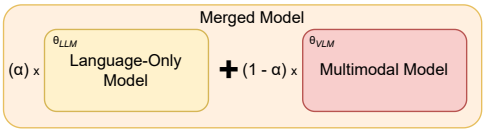


Figure 1: Weighted merging of language-only and multimodal models in the form of linear interpolation.

Despite the general good performance of MLMs in multimodal tasks, previous work shows that these models tend to underperform in *language-only* tasks (Zhuang et al., 2024). Recent work in the multimodal BabyLM challenge also points to the same issue (Amariucai and Warstadt, 2023; Klerings et al., 2024). Therefore, our aim in this paper is to first test our own models on multimodal and text-only benchmarks, and second, if we observe the same issue, to try to mitigate it.

We develop language-only and multimodal models, the latter of which outperforms previous BabyLM baselines.[1] However, our results indeed confirm that our developmentally plausible MLMs lack in text-only benchmarks. Hence, we explore a model augmentation technique to potentially overcome this shortcoming: **model merging**. Model merging has been utilized to prevent catastrophic forgetting and combine the capabilities of multiple models trained on different tasks, datasets, or modalities (Yang et al., 2024; Dash et al., 2025).

In our approach, during inference time, we fuse the parameters of models trained on text-only and multimodal data in a straightforward, training-free way (see Figure 1). Our results indicate that such an augmentation yields a single model maintaining accuracy and robustness in both text-only and multimodal benchmarks in the earlier and later stages of training the multimodal model.

[1]Code and models available at https://github.com/ecekt/babylm_multimodal_model_merging

Proceedings of the First BabyLM Workshop, pages 66–75
November 8, 2025

2 Background

We first go into detail about the multimodal approaches to the BabyLM challenge in Section 2.1, and then, provide a summary of the recent work on model merging in Section 2.2.

2.1 Developmentally Plausible Multimodal Language Models

The BabyLM initiative encourages the development of models that can be small-scale, trained on smaller sets of data, using various techniques such as model compression, learning from interaction, knowledge distillation. Our focus is on the **multimodal track**. Multimodal models have been explored in the 2nd BabyLM challenge (Choshen et al., 2024) and re-introduced in the 3rd BabyLM challenge (Charpentier et al., 2025) as the submitted models did not outperform the baselines released by the BabyLM organizers (Hu et al., 2024). The baselines were GIT (Wang et al., 2022) and Flamingo (Alayrac et al., 2022) models trained on the BabyLM's multimodal corpus to ground language to vision.

These models encode image inputs and generate text using text decoders conditioned on visual tokens. The methodologies from the past submissions include curriculum learning where the captions were ordered based on the number of concepts they included (Saha et al., 2024), where this helped on developmentally plausible benchmarks. Pretraining on text also seems to be beneficial; Saha et al. (2024) investigates first training on text and then captions along with curriculum learning using image-caption pairs. However, in general, it appears to be difficult to observe a strong pattern across model types, datasets and tasks.

Another work reports a related result where learning in phases appears to benefit multimodal BabyLM models (AlKhamissi et al., 2024). The model first learns the language-only tasks, then grounding, followed by self-synthesized data and more advanced reasoning tasks.

There are also contributions to the language-only track where the models were influenced or informed by multimodal input (Fields et al., 2023; Amariucai and Warstadt, 2023).

Klerings et al. (2024) explore a weighted loss function for text-only and multimodal data during training. However, they show that vision does not significantly benefit the performance in language-only benchmarks. This is in line with prior findings showing limited or no improvements when incorporating visual data (Amariucai and Warstadt, 2023; Zhuang et al., 2024), with the exception of low-data regimes (Zhuang et al., 2024), which inspired our work.

2.2 Model Merging

Model merging has been utilized as a technique for adaptively extending the capabilities of models or balancing performance during inference time in the tasks multiple models were trained on. See (Yang et al., 2024) and (Goddard et al., 2024) for surveys of various merging techniques.

A straightforward averaging technique called 'model soups' has been found beneficial in improving accuracy and robustness. The techniques involve combining the parameters of multiple models trained on different hyperparameters, in addition to more sophisticated weighted merging methods (Wortsman et al., 2022; Matena and Raffel, 2022). Similarly, Aakanksha et al. (2024) find that merging models is better than mixing training data for facilitating safety and multilingual generalizability.

Regarding vision-and-language models, Zhu et al. (2025) learn modules for various multimodal tasks that are later merged; whereas (Li et al., 2025) exploit text-only reward models to transfer to vision-and-language reward models in a cross-modal model merging scheme.

Closer to our approach, AyaVision is an example of cross-modal merging to maintain text-only capabilities within multimodal models to prevent catastrophic forgetting (Dash et al., 2025). The authors built their multimodal model on their best-performing text-only checkpoint, which makes the setup more suitable for merging. Similarly, Sung et al. (2023) conduct detailed experiments on multimodal model merging, finding that simple linear interpolation is a competitive and efficient method, which we also opt for in this work to test its effectiveness in low-data and low-compute settings.

3 Data

To train our models, we use the data from the multimodal BabyLM challenge, which consists of 2 parts: text-only and multimodal.

Text-only. We use the 50M-word text data provided by the BabyLM challenge. This data consists of text stemming from 6 sources as explained by Choshen et al. (2024), and corresponds to the

Data	Train	Val
Localized Narratives	729349	38387
CC3M	2061837	108518
BabyLM - text-only	5492930	289102
Total	8284116	436007

Table 1: Number of samples per dataset in the splits we created (after filtering, validating and deduplicating CC3M images and captions, and trimming it to fit 100M words in total).

first halves of the text-only subsets released for the language-only challenge of the BabyLM task.

Multimodal. This part includes image-caption pairs from Localized Narratives (Pont-Tuset et al., 2020) and Conceptual Captions 3M (Sharma et al., 2018). We download the Localized Narratives (LN) images and captions from the dataset's website.[2] We use the COCO (train) (Lin et al., 2014) and Open Images (train and test) (Kuznetsova et al., 2020) subsets of LN.[3]

Additionally, we download the captions for the existing images from the Conceptual Captions (CC3M) dataset.[4] Filtering out the images that do not exist anymore as well as the corrupt and duplicate image files, we end up with fewer than 3M images, which is lower than the provided captions for the previous multimodal BabyLM challenge.

Final dataset. The statistics of our final dataset are provided in Table 1, with our random train and validation splits where 95% of each subset contributes to the training set and the rest goes into the validation set.

4 Methodology

Models. We modify the implementation of LLaVa (Liu et al., 2023, 2024) from HuggingFace[5], inheriting the LlavaForConditionalGeneration model, and replace the visual encoder with DINOv2-large (Oquab et al., 2024)[6]. We also make necessary changes to the image processing code and modeling code in relation to the dimensions of the image features. We randomly initialize a 6-layer version of this model as the language model, together with a mapping layer that projects the image representations to the language model's space.

Image representations. Unlike the BabyLM benchmark's image representations from last year that are 768-dimensional vectors from DINOv2 ViT-Base, we use the large version of DINOv2, which processes images into 256 image tokens of 1024 dimensions. While we originally intended to feed all 256 image tokens extracted from the vision encoder, due to time and compute constraints, we modified the model to feed a single pooled image token directly. We implemented a version of the model where we pre-extract all image token representations and mean-pool them. This single summary token (1024 dim) is fed to the LM directly (bypassing the vision tower). The summary image representation goes through a multimodal projector composed of a linear layer projecting from 1024 to 768 dimensions, GeLU activation and another linear layer projecting from 768 to 768. This multimodal projector is trained along with the language model, while keeping the image representations frozen.

For text-only data, we create a black image (640 x 420) and always input the features of this placeholder image both in the text-only model and the multimodal model.

Training the tokenizer. Using all the text in the final dataset (token count = 100M), we train a tokenizer from scratch employing the configurations of the LLaVA tokenizer (LlamaTokenizerFast, a byte-pair encoding model based on SentencePiece), with a vocabulary size of 30000 including a special token for image representations. Using the BERT pre-tokenizer, we apply splitting on whitespace and punctuation. This preprocessing yields a 1.36 word-to-subword ratio.[7]

Intermediate checkpoints. To investigate the learning speed and model behavior dynamics, we save checkpoints (every 1M words up to 10M, every 10M up to 100M, every 100M up to 1B). We estimate the words-seen using the ratio of word-pieces to actual words in our dataset (1.36). We use this ratio to roughly determine how many 'words'

[2] https://google.github.io/localized-narratives/

[3] Although the BabyLM OSF repository at https://osf.io/ad7qg/files provides captions and extracted image representations for this subset, we noticed a discrepancy in the number of samples compared to the original LN. It seems that the test set of the Open Images subset is also counted in the BabyLM corpus to end up with the 50M word count. Therefore, we also included that part. Localized Narrative subset IDs of Open Image downloaded from https://storage.googleapis.com/openimages/web/download_v7.html. COCO images from https://cocodataset.org/

[4] We download the captions from https://ai.google.com/research/ConceptualCaptions/download and the images using the script provided at https://github.com/igorbrigadir/DownloadConceptualCaptions.

[5] 'llava-hf/llava-1.5-7b-hf'

[6] 'facebook/dinov2-large'

[7] Words: 99,999,990. Subwords (as tokenized by our tokenizer, skipping special tokens): 136,034,832.

the models were exposed to in the training batches (excluding special tokens).

Hyperparameters and setup. We follow the restrictions of the BabyLM challenge, 100M words, 10 epochs, resulting in 1B tokens seen in total.

We set the maximum length to 150 tokens. We truncate longer samples if they do not fit this constraint; if they are shorter, we pad them.

We train the models on 2 A10 GPUs with 24GB memory on CrossEntropy Loss using the AdamW optimizer with a learning rate of 1e-4. For the multimodal model, batches can contain text-only or multimodal data, and the loss is calculated in the same way for both modalities. We use fp16 half precision and make use of the Accelerate library for data parallelism to speed up training. We accumulate gradients for 8 steps and then apply gradient updates to optimize the model, effectively increasing our batch sizes from 64 to 512. The numbers of words seen are gathered from each GPU and only logged and checked in the main process. We opted for a smaller layer number (6) to allow for a speed-up in the training by exposing the model to larger and more batches in a shorter amount of time. It takes 6.7 days for the multimodal model (with text-only and multimodal data) to be trained, and the text-only model 4.3 days.

5 Model Merging at Inference Time

Since our language-only and multimodal models share the same architecture, random initialization and the text-only data, they can be combined in a straightforward way. We apply a simple weighted sum of the multimodal model's parameters and the text-only model's parameters. We experiment with merging weights α of 0.3, 0.5, and 0.8,[8] where θ indicates all the trainable parameters of a model:

$$\theta_{merged} = \alpha\theta_{LLM} + (1-\alpha)\theta_{VLM} \quad (1)$$

In this way, the merged model is a linear interpolation of the multimodal and text-only models.

6 Benchmarks

We modify the evaluation pipeline provided by the BabyLM challenge[9] to run zero-shot evaluations across our checkpoints using the benchmarks provided. For Winoground, we write our own evaluation code.

Language-only benchmarks. We run the evaluation pipeline for all the tasks in BLiMP (Warstadt et al., 2020), EWoK, entity tracking (assign the highest probability to the correct continuation) (Kim and Schuster, 2023), Wug past tense (Weissweiler et al., 2023), wug adjective nominalization (Hofmann et al., 2025) testing morphological capabilities (correlating model probabilities to human judgments). BLiMP and BLiMP-supplement (more challenging samples) evaluate whether the models capture grammatical phenomena, where one grammatical and one ungrammatical sentence are pitted against each other, testing models' capabilities related to syntax, morphology and semantics. EWoK (Ivanova et al., 2025) focuses on world knowledge and reasoning about e.g., social, physical, spatial relations.

Multimodal benchmark. We experiment on Winoground (Thrush et al., 2022), where pairs of images with very similar captions are provided. Winoground consists of 400 samples, where each sample has 2 images and 2 captions. These 2 captions have the same words, but in different orders to match the image contents (see an example in Figure 2).

Figure 2: An example from Winoground. *Left:* 'painting the white wall red'. *Right:* 'painting the red wall white'.

We input the image and 2 captions separately to the model to obtain predictions for Winoground. If the likelihood of the correct caption is higher, we increase the accuracy. We use the unpaired text-score as used in previous BabyLM work, where we consider each image-caption pair separately. We use the full Winoground dataset available on HuggingFace, unlike the filtered version in the BabyLM evaluation suite.

Winoground tests abilities requiring compositionality, sensitivity to word order, commonsense reasoning, pragmatics and overall more fine-grained visual and linguistic analyses involving unusual images and texts (Diwan et al., 2022).

[8] The weights were chosen to reflect equal contribution from both models (0.5) and a skewed contribution from one model (0.3–more VLM and 0.8–more LLM).

[9] https://github.com/babylm/evaluation-pipeline-2025

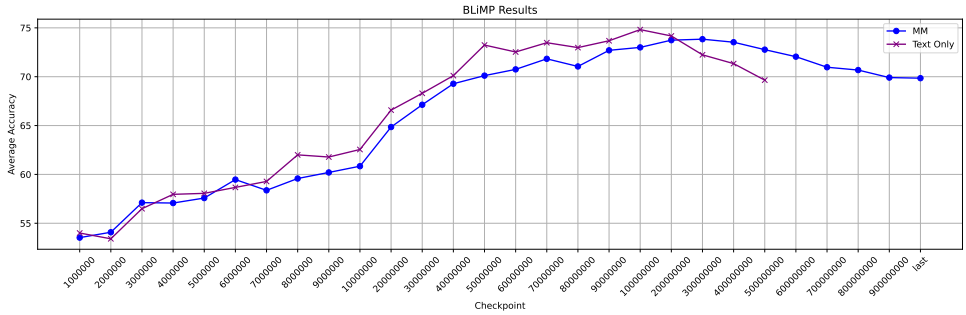


Figure 3: Average accuracies for the text-only model and the multimodal model over the training checkpoints, for the BLiMP **full** benchmark.

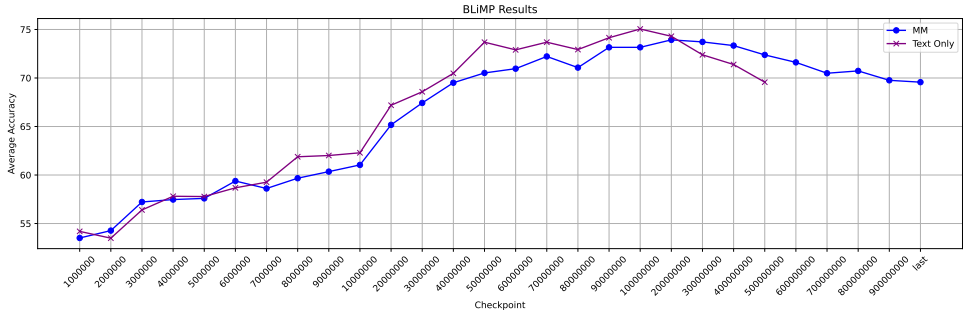


Figure 4: Average accuracies for the text-only model and the multimodal model over the training checkpoints, for the BLiMP **fast** benchmark.

Winoground is a difficult dataset, with previous BabyLM work yielding accuracies as follows: 2024 baselines Flamingo: 51.6 GIT: 55.5; 2025 baselines Flamingo: 54.8, GIT, 56.2.[10]

7 Results

Results on benchmarks. We first obtain the results on the full BLiMP evaluation, which is reported in Figure 3. Our best language-only model reaches 74.82 accuracy at the 100M checkpoint. **Our best multimodal model yields 73.84 accuracy, surpassing the multimodal BabyLM baselines as well as the current 2 submissions on the multimodal BabyLM leaderboard** (2024 Flamingo: 70.9, GIT: 65.2, 2025 Flamingo: 70.9, GIT: 72.2). We see that, generally, the text-only model outperforms or is on par with the multimodal model, except some later checkpoints.

We use the 'fast' versions of the benchmarks that contain a smaller set of samples to obtain the following results due to time and compute constraints.[11] In Figure 4, we depict the performance of the text-only and multimodal model checkpoints on the BLiMP fast benchmark, which yields outcomes closely aligned with those obtained from the full benchmark.

We see similar trends for BLiMP supplement (Figure 5), Wug past tense (Figure 6) and adjective nominalization (Figure 7) benchmarks. This is in line with previous work indicating that multimodal data tend not to benefit performance on language-only benchmarks.

When we look at the results on the Entity Tracking (Figure 8) and EWoK benchmarks (Figure 9), however, we see trends where multimodal checkpoints clearly outperform the text-only checkpoints. This could be due to the focus of these datasets, which is more knowledge- and semantics-oriented rather than grammatical, therefore, the multimodal data such as the image descriptions in narrative form from the Localized Narratives dataset could have helped.

Although our models do not perform well in the BLiMP supplement and Wug past tense benchmarks, they show competitive performance in the remaining tasks.

Results on model merging. We use BLiMP as a

[10] 2024 baselines from: https://github.com/babylm/evaluation-pipeline-2024/, 2025 baselines from: https://huggingface.co/spaces/BabyLM-community/babylm-leaderboard-2025-all-tasks

[11] We noticed that the Wug fast and full benchmarks are in fact identical.

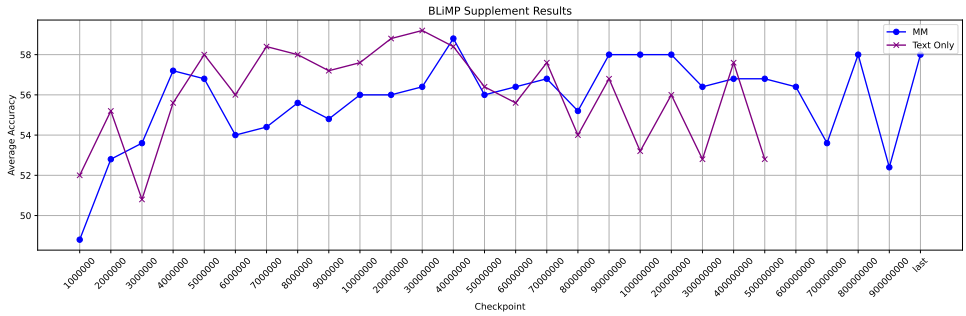


Figure 5: Average accuracies for the text-only model and the multimodal model over the training checkpoints, for the BLiMP supplement fast benchmark.

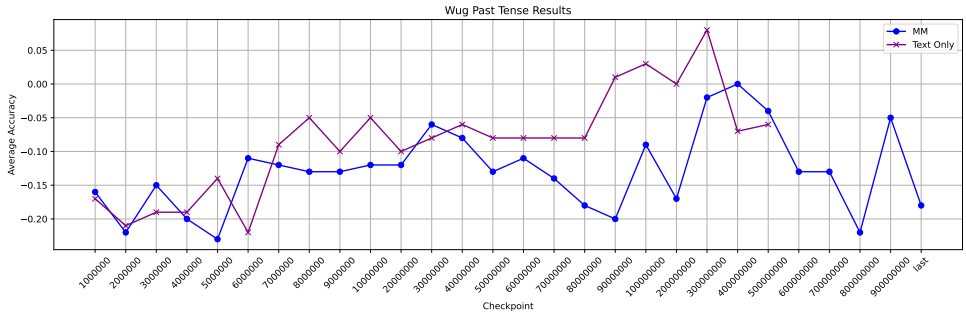


Figure 6: Correlation between model predictions and human responses from the Wug past tense benchmark, for the text-only model and the multimodal model over the training checkpoints.

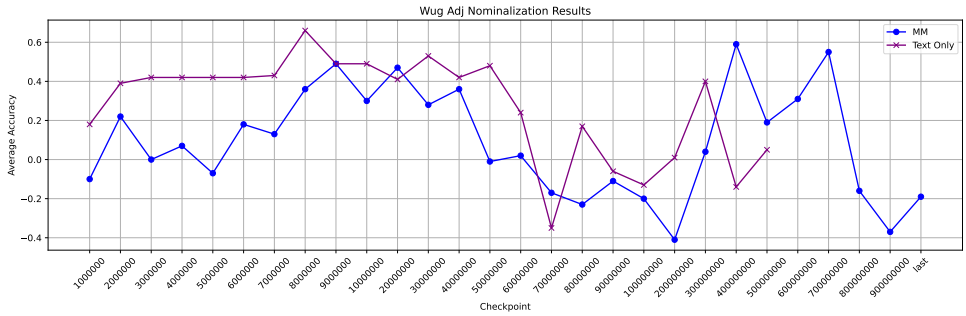


Figure 7: Correlation between model predictions and human responses from the Wug adjective nominalization benchmark, for the text-only model and the multimodal model over the training checkpoints.

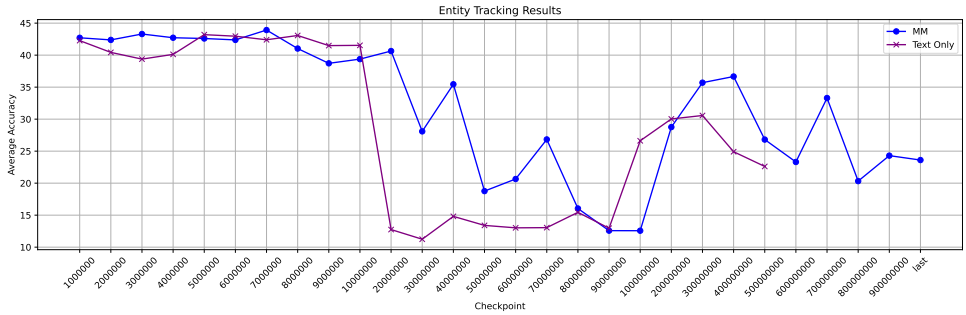


Figure 8: Average accuracies for the text-only model and the multimodal model over the training checkpoints, for the Entity Tracking fast benchmark.

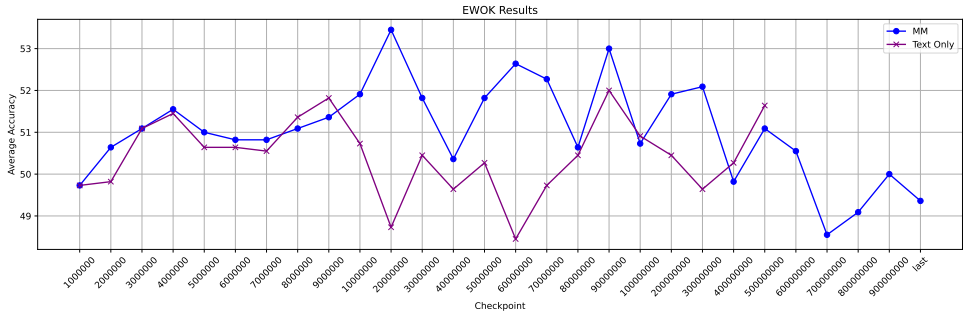


Figure 9: Average accuracies for the text-only model and the multimodal model over the training checkpoints, for the EWoK fast benchmark.

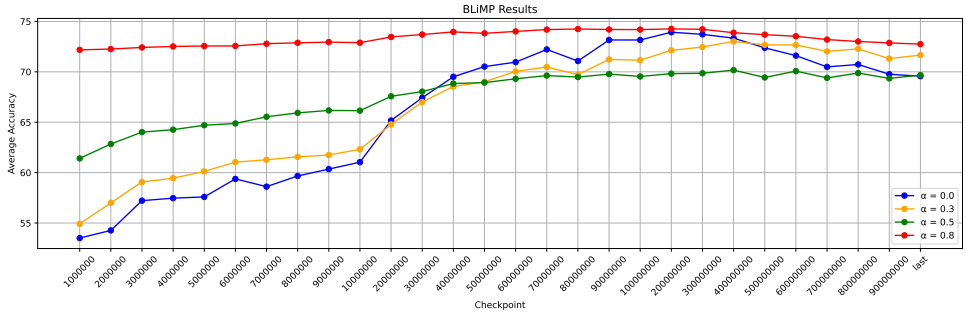


Figure 10: Average accuracies for the merged models with different weights (higher α indicates more contributions from the language-only model), along with the training dynamics of the multimodal model for the BLiMP fast benchmark.

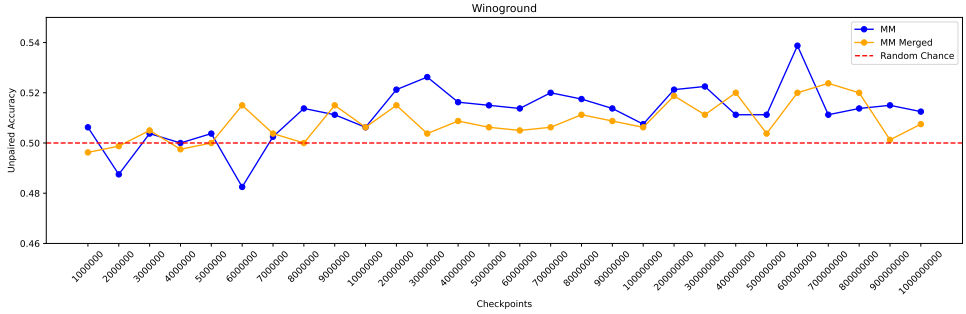


Figure 11: Average accuracies for Winoground. MM represents the multimodal model checkpoints and MM Merged indicates a merged model with $\alpha = 0.3$, using the language-only checkpoint with the highest BLiMP score. The red dotted line indicates random chance accuracy. The top score in the leaderboard on the filtered version is 56.2.

use case for our model merging experiments. We merge each multimodal checkpoint model with the language-only model that performs best in BLiMP (100M checkpoint) with varying weights. Figure 10 illustrates the model-merging results when combining the language-only and multimodal models using $\alpha = 0.3$, 0.5 and 0.8 as the weights of the language-only model and $1 - \alpha$ for the multimodal model. Merging the trained language-only model in the early training stages of the multimodal model meaningfully helps in getting better results in BLiMP. Additionally, in the later checkpoints when the multimodal model's language-only capabilities begin to drop, 0.3/0.7 merging scheme helps the model maintain language-only capabilities.

To check whether merging with the language-only model affects multimodal performance, we also look at the accuracy on the Winoground benchmark after merging models. The results for Winoground are provided in Figure 11, showing that in some checkpoints, merging can actually be beneficial without significantly decreasing multimodal scores.

8 Conclusion

We have investigated whether model modification in the form of model merging at inference time would benefit multimodal BabyLM models in language-only and multimodal tasks. Our results showed that, indeed, multimodal models tend to underperform in text-only benchmarks that focus on grammar (although surpassing previous baselines) and model merging with text-only models can help alleviate this issue to some extent. Future work can explore other model merging techniques and the effects of model merging in other benchmarks.

Limitations

Due to time and compute constraints, we altered our intended initial setup where the model is fed 256 image patches into one where a single, pooled image representation is relayed to the model. This might cause information loss and performance drop, and ideally, we would like to provide the whole set of image patches. We tested models with 6 transformer layers, which is quite few compared to state-of-the-art models. Therefore, this might have resulted in lesser performance. However, we believe that our results shed light on what to expect in compute and data-efficient/scarce setups, which should be investigated further using more seeds and different training orders for robustness and generalizability of the conclusions. Additionally, although the set of benchmarks we tested on does not cover the full spectrum of language-only and multimodal tasks in the BabyLM challenge, we think that they span a reasonable range of them, providing insights into the dynamics visuo-linguistic processes as training progresses.

Acknowledgments

The research reported in this paper was supported by the European Research Council (ERC), grant 101088098 - MEMLANG. The first author utilized compute resources granted by the Dutch Research Council (NWO). The second author acknowledges support from NWO, Open Competition XS project number 406.XS.25.01.104. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

Aakanksha, Arash Ahmadian, Seraphina Goldfarb-Tarrant, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. 2024. Mix data or merge models? optimizing for performance and safety in multilingual contexts. In *Neurips Safe Generative AI Workshop 2024*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*.

Badr AlKhamissi, Yingtian Tang, Abdülkadir Gökce, Johannes Mehrer, and Martin Schrimpf. 2024. Dreaming out loud: A self-synthesis approach for training vision-language models with developmentally plausible data. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 244–251, Miami, FL, USA. Association for Computational Linguistics.

Theodor Amariucai and Alexander Scott Warstadt. 2023. Acquiring linguistic knowledge from multimodal input. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 128–141, Singapore. Association for Computational Linguistics.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [call for papers] the 2nd babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2404.06214.

Saurabh Dash, Yiyang Nan, John Dang, Arash Ahmadian, Shivalika Singh, Madeline Smith, Bharat Venkitesh, Vlad Shmyhlo, Viraat Aryabumi, Walter Beller-Morales, Jeremy Pekmez, Jason Ozuzu, Pierre Richemond, Acyr Locatelli, Nick Frosst, Phil Blunsom, Aidan Gomez, Ivan Zhang, Marzieh Fadaee, and 6 others. 2025. Aya vision: Advancing the frontier of multilingual multimodality. *Preprint*, arXiv:2505.08751.

Anuj Diwan, Layne Berry, Eunsol Choi, David Harwath, and Kyle Mahowald. 2022. Why is winoground hard? investigating failures in visuolinguistic compositionality. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Clayton Fields, Osama Natouf, Andrew McMains, Catherine Henry, and Casey Kennington. 2023. Tiny language models enriched with multimodal knowledge from multiplex networks. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 47–57, Singapore. Association for Computational Linguistics.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. Association for Computational Linguistics.

Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *Preprint*, arXiv:2405.09605.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Alina Klerings, Christian Bartelt, and Aaron Mueller. 2024. Developmentally plausible multimodal language models are highly modular. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 118–139, Miami, FL, USA. Association for Computational Linguistics.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. 2020. The open images dataset V4. *International journal of computer vision*, 128(7):1956–1981.

Chen-An Li, Tzu-Han Lin, Yun-Nung Chen, and Hung-yi Lee. 2025. Transferring textual preferences to vision-language understanding through model merging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 923–943, Vienna, Austria. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Michael Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, and 7 others. 2024. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. Featured Certification.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *ECCV*.

Rohan Saha, Abrar Fahim, Alona Fyshe, and Alex Murphy. 2024. Exploring curriculum learning for vision-language tasks: A study on small-scale multimodal training. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 65–81, Miami, FL, USA. Association for Computational Linguistics.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Yi-Lin Sung, Linjie Li, Kevin Lin, Zhe Gan, Mohit Bansal, and Lijuan Wang. 2023. An empirical study of multimodal model merging. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1563–1575, Singapore. Association for Computational Linguistics.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. GIT: A generative image-to-text transformer for vision and language. *Transactions on Machine Learning Research*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *Preprint*, arXiv:2408.07666.

Didi Zhu, Yibing Song, Tao Shen, Ziyu Zhao, Jinluan Yang, Min Zhang, and Chao Wu. 2025. REMEDY: Recipe merging dynamics in large vision-language models. In *The Thirteenth International Conference on Learning Representations*.

Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. 2024. Visual grounding helps learn word meanings in low-data regimes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1311–1329, Mexico City, Mexico. Association for Computational Linguistics.

TafBERTa: Learning Grammatical Rules from Small-Scale Language Acquisition Data in Hebrew

Anita Gelboim[1], **Elior Sulem**[1,2]
[1]Faculty of Computer and Information Science, Institute for Applied AI Research
[2]The School of Brain Sciences and Cognition
Ben-Gurion University of the Negev
anitana@post.bgu.ac.il, eliorsu@bgu.ac.il

Abstract

We present TafBERTa, a compact RoBERTa (Liu et al., 2019) based language model tailored for Hebrew child-directed speech (CDS). This work builds upon the BabyBERTa (Huebner et al., 2021) framework to address data scarcity and morphological complexity in Hebrew. Focusing on determiner-noun grammatical agreement phenomena, we show that TafBERTa achieves competitive performance compared to large-scale Hebrew language models while requiring significantly less data and computational resources. As part of this work, we also introduce a new corpus of Hebrew CDS, HTBerman, aligned with morphological metadata and our new grammatical evaluation benchmark for Hebrew, HeCLiMP, based on minimal pairs. Our results demonstrate the effectiveness of TafBERTa in grammaticality judgments and its potential for efficient NLP in low-resource settings.

1 Introduction

In the last few years, Language Models (LMs) have expanded in both parameter count and training data size (Kaplan et al., 2020). Besides the numerous contributions to NLP tasks (Min et al., 2023; Zhao et al., 2023) and their application in many domains (Chiarello et al., 2024), this trend brings various challenges, including computational inefficiency, increased environmental costs and difficulties in adapting models to low-resource languages.

Recently, works such as BabyBERTa (Huebner et al., 2021) and the BabyLM Challenge (Warstadt et al., 2023) addressed these aspects by developing English compact models trained on child-directed language, demonstrating strong grammatical abilities with minimal data. However, no such efforts have not been done in Hebrew, a low resource language where data scarcity is a main challenge, leaving a significant gap in efficient, accessible language modeling for Hebrew NLP.

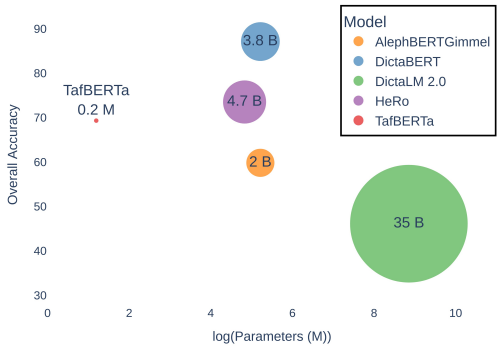


Figure 1: Overall accuracy of Hebrew language models on the HeCLiMP benchmark (see Section 4). Bubble size represents the number of words seen during training, while the x-axis indicates the logarithm of model parameters (M).

In this paper, we introduce TafBERTa, a compact RoBERTa (Liu et al., 2019) based model optimized for Hebrew. To assess the effectiveness and efficiency of TafBERTa, we pose several key research questions. First, we investigate how TafBERTa's smaller size—defined by both its reduced number of parameters and the smaller dataset used for training—impacts its performance relative to HeRo (Shalumov and Haskey, 2023, a Hebrew version of RoBERTa). This comparison assesses whether a more compact architecture can achieve competitive results, despite having fewer computational resources and less training data (Q1). Beyond this direct comparison, we explore whether a search over the parameter space was necessary for optimizing TafBERTa's performance, particularly in training a RoBERTa architecture on the HTBerman child-directed speech corpus we introduce (Q2). Additionally, we evaluate the capabilities of TafBERTa against other Hebrew models using other architectures or tokenization methods, to establish its relative strengths and weaknesses within

Proceedings of the First BabyLM Workshop, pages 76–90
November 8, 2025

the Hebrew NLP landscape (Q3). Finally, we assess the adaptability of TafBERTa's architecture by testing its ability to learn from alternative data sources, specifically evaluating its performance when trained on Wikipedia-derived Hebrew text rather than child-directed speech (Q4). These questions guide our evaluation, providing insights into both the efficiency of small-scale models and the nuances of Hebrew NLP.

Our contributions: (1) introducing TafBERTa, an efficient Hebrew model, (2) introducing HTBerman dataset for Hebrew Child-Directed Speech (CDS), (3) presenting HeCLiMP, a benchmark for Hebrew grammatical evaluation tailored to CDS, and (4) conducting a comparative study against HeRo and other models. Results show TafBERTa achieves competitive performance despite its reduced size, highlighting the potential of small, well-tuned models for low-resource NLP.[1]

2 Related Work

2.1 Baby Language Models

In response to the parameter and data expansion in Large Language Models (LLMs), research has increasingly turned toward smaller, more efficient models that retain strong linguistic capabilities. The BabyLM challenge (Warstadt et al., 2023) exemplifies this shift, encouraging the development of compact models that learn from limited yet high-quality data, mimicking human language acquisition. A key resource is the CHILDES database (MacWhinney, 2000), which includes well-established corpora of casual speech to children that has shaped studies in cognitive linguistics and NLP (Huebner and Willits, 2021; Mueller and Linzen, 2023). Building on this foundation, BabyBERTa (Huebner et al., 2021) was introduced as a scaled-down RoBERTa variant trained on child-directed language, demonstrating that even with fewer parameters and less training data, models can develop strong grammatical abilities. Evaluation of such models relies on syntactic and grammatical benchmarks like BLiMP (Warstadt et al., 2020) and Zorro (Huebner et al., 2021), which test linguistic phenomena.

We address here these questions from the perspective of the Hebrew language, tackling challenges in low-resource language adaptation.

[1]We release the code and datasets at https://github.com/NLU-BGU/tafberta/ to facilitate reproducibility and future research.

2.2 Baby Language Models in Other Languages

While much of the research on baby language models has focused on English, recent work has expanded these efforts to additional languages.
For instance, in Italian, Capone et al. (2024) introduced a benchmark designed for the standardized evaluation of Italian BabyLMs. To assess its effectiveness, researchers applied the benchmark to Minerva (Orlando et al., 2024), an LLM pretrained from scratch on Italian. The results revealed that Minerva struggled with certain linguistic aspects, achieving an age-equivalent score of just four years. This under-performance highlights the necessity of refining model training approaches to improve language acquisition efficiency. In German, Bunzeck et al. (2025) studied the effect of utterance-level construction distributions in German child-directed and child-available speech on the model performance at the word-level, syntactic and semantic levels. The grammatical abilities of Baby Language Models beyond English have also been investigated in Salhan et al. (2024), covering Chinese, French, German, and Japanese, and focusing on the effect of curriculum learning. Focusing on phonology, Goriely and Buttery (2025) trained small monolingual language models on child-directed and child-produced speech, covering 11 languages.

Several recent studies have explored second language acquisition (L2) with language models, drawing parallels to human language learning processes. In Italian, BAMBINO-LM (Shen et al., 2024), a bilingual pre-training approach for BabyLM, enhances Italian proficiency while maintaining English skills, using alternation and PPO (proximal policy optimization)-based perplexity rewards. Yadavalli et al. (2023) and Oba et al. (2023) examined L2 acquisition in neural models, by pretraining LMs in a certain language, further training them in English as an L2, and evaluating and analyzing their linguistic generalization in L2. They found that L1 pretraining accelerates L2 learning, with varying linguistic transfer effects.

We focus here on Hebrew, a low-resource language for which Baby Language Models have not been explored, and address it in a monolingual setting.

2.3 Hebrew Language Models

Hebrew language models continue to lag behind their English counterparts, facing challenges in data

availability and computational efficiency (Tsarfaty et al., 2019).

Compared to English, Hebrew has more limited corpora for training large-scale models, making it difficult to achieve the same level of performance. Despite this limitation, several Hebrew language models have been developed to bridge the gap. AlephBERT (Seker et al., 2022), AlephBERTGimmel (Gueta et al., 2023) and HeRo (Shalumov and Haskey, 2023) were among the first transformer-based models for Hebrew providing contextual embeddings suited to the language's structure. DictaBERT (Shmidman et al., 2023) and its successor, DictaLM 2.0 (Shmidman et al., 2024), further refined Hebrew language modeling, improving general-purpose NLP tasks. While these advancements mark progress, Hebrew NLP still requires larger, higher-quality datasets and more efficient training strategies to reach the capabilities of English LLMs.

Another difficulty in Hebrew NLP is the linguistic challenge (Tsarfaty et al., 2019). Hebrew is a morphologically-rich language (MRL), and in MRLs, every input token could contain some lexical and functional units, known as morphemes, each playing a distinct role in shaping the syntactic or semantic representation. One challenge arises from the necessity to segment Hebrew tokens into their constituent morphemes before processing Hebrew texts. The segmentation process has experienced significant advancement with the utilization of tools like YAP (Yet Another (Natural Language) Parser, More et al., 2019) or the DictaBERT model (Shmidman et al., 2023), which has been fine-tuned specifically for the segmentation task.

In Hebrew NLP, only after performing the segmentation phase, we should chose the tokenizer. The most popular tokenization algorithms are Byte-Pair Encoding (BPE) (Sennrich et al., 2016) and Google's WordPiece (Song et al., 2021), which are used by RoBERTa and BERT respectively. Another method based on morphemes is used by HeBERT and AlephBERTGimmel. See Gazit et al. (2025) and Gorman and Pinter (2025) for further perspectives on Hebrew tokenization.

Our work takes these challenges into account by focusing on data efficiency and morphological complexity, designing a model that learns from limited yet high-quality Hebrew data while addressing the constraints of low-resource language modeling.

2.4 Probing Grammatical Rule Learning

Recent LLMs have demonstrated remarkable success in addressing a wide array of downstream tasks. However, there is still a need to determine the extent to which these LLMs comprehend the syntax of natural languages. To tackle this question, several studies have examined the syntactic understanding of language models using tailored datasets specifically designed for targeted syntactic evaluations. One way to examine it is using a probing task i.e., a classification problem that focuses on simple linguistic properties of sentences (Conneau et al., 2018). The objective of this task is to assess the quality of a model, focusing on its language proficiency, particularly in syntax and grammar. Some explored this question by evaluating language models' (LMs) preferences between *minimal pairs* (MP) of sentences differing in grammatical acceptability, as in the next example:

1. Imagination is more important than knowledge. (grammatical)

2. Imagination are more important than knowledge. (ungrammatical)

A MP is classified correctly if a LM assigns a higher probability to the grammatical sentence than to the ungrammatical one.
The Benchmark of Linguistic Minimal Pairs (BLiMP, Warstadt et al., 2020) is a benchmark designed with linguistic principles in mind. It evaluates the ability of language models to discern acceptability differences across various English phenomena. However, most of the studies have focused on English and other European languages. Only few studies extended this investigation to non-European languages, such as CLIMP (Xiang et al., 2021) and JBLiMP (Someya and Oseki, 2023), for Chinese and Japanese languages respectively. The authors of CLIMP built the corpus of Chinese MPs in the by generating data from grammar templates for every paradigm they incorporate, building an annotated vocabulary, and generating sentences by sampling words from the vocabulary, which is a translation of BLIMP English Vocabulary. The authors of JBLIMP created the corpus of Japanese MPs based on acceptability judgments extracted from journal articles in theoretical linguistics. These minimal pairs are grouped into 11 categories, each covering a different linguistic phenomenon. In some other languages (specifically Italian, English, Hebrew and Russian), Gulordava

et al. (2018) strengthen the evaluation paradigm of MPs in terms of subject-verb agreement. Their assessment involves nonsensical sentences, challenging language models by eliminating reliance on semantic or lexical cues ("The colorless green ideas I ate with the chair sleep furiously"). The evaluation test sets are extended to other phenomena, resulting in the CLAMS benchmark (Mueller et al., 2020).

Differently from the Hebrew section of CLAMS, we build here a grammatical benchmark (HeCLiMP) tailored to CDS, abstracting away from lexical complexity, yet addressing two main Hebrew grammatical phenomena that exemplify the rich morphology in Hebrew. HeCLiMP also differs from CLAMS by being constructed directly in Hebrew, abstracting away from the effects of translation from the English language.

Grammatical benchmarks in English that are tailored to CDS include Zorro (Huebner et al., 2021) and BabySLM (Lavechin et al., 2023).

3 Training Data: HTBerman dataset

Our main corpora of interest are the original version of CHILDES Hebrew Berman Longitudinal Corpus (Armon-Lotem, 1996, Berman corpus)[2], written in latin-based phonemic Hebrew talk transcription and a version of it written in standard Hebrew script (Albert et al., 2012).[3]

The Berman corpus comprises longitudinal naturalistic data gathered weekly from four Hebrew-speaking children. In order to fairly compare with other Hebrew language models, we use the version of the corpus written in standard Hebrew script. Since the latter does not contain the metadata present in the original version, we merge the two versions, creating a comprehensive dataset that incorporates Hebrew text along with all the annotations, at the utterance and word levels.

As part of the corpora merge, we performed data cleaning, which included morphological segmentation (More et al., 2019) and punctuation correction (See Section A for the details). The resulted corpus **HTBerman** (Hebrew Transcription Berman) contains 53K sentences, 233k words and ∼8K unique words of Hebrew transcribed CDS.

[2]This corpus is part of CHILDES project (MacWhinney, 2000).

[3]We use as initial data the outputs of the automatic converter built by Albert et al. (2012).

3.1 Corpora

3.1.1 CHILDES Hebrew Berman Longitudinal Corpus

Our main corpus of interest is CHILDES Hebrew Berman Longitudinal Corpus. This corpus is transcribed with a Latin-based phonemic of Hebrew talks. The transcripts were all transcribed in the CHAT format (CHILDES) with adaptations to Hebrew. The dataset comprises longitudinal naturalistic data gathered weekly from four Hebrew-speaking children. These children are all native Hebrew speakers raised in households where Hebrew is the primary language and the environment is characterized by high levels of education. Each child was audio-recorded in various settings at their home, including mealtime, bath time, solitary play and interactions with siblings, parents and grandparents. This corpus includes the following morphological annotations:

- **Participants:** This component refers to the individuals involved in the conversation. In CHILDES, the convention is to designate the child being studied as CHI and the child's mother as MOT. Each utterance in the conversation begins with an indication of the participant speaking, denoted by an asterisk (*) followed by the participant code.
- **Transcriptions:** Transcriptions capture the spoken language in written form.
- **Dependent tiers:** These are additional layers of linguistic information associated with each transcription line. They are preceded by a percentage symbol (%) and are linked to the transcription line immediately above. Dependent tiers can include morphological information (%mor), grammatical relations (%gra), intonation (%int) and others. While some tiers are common in CHILDES datasets, none are obligatory.
- **The %mor tier:** This tier provides morphological information about each word in the transcription. It aligns one-to-one with the segmented words and disregards any annotations present in the transcription line. Each item in the %mor tier consists of a part-of-speech tag followed by inflectional or derivational information, separated by a pipe (|). For example, "qn|more" indicates a nominal quantifier aligned with the word "more".

- **The %gra tier:** The grammatical relations tier represents relationships between words in terms of heads and dependents in dependency grammar. Each item in the %gra tier corresponds one-to-one with the segmented words in the transcription, as well as with items in the %mor tier. It specifies the syntactic relationship between words, such as subject-verb or quantifier-noun.

- **Other tiers:** In addition to %mor and %gra, there may be other dependent tiers providing further linguistic or contextual information. For example, the %int tier captures intonation patterns, while others may contain information about the recording session or the context of the conversation.

We accessed the data using PyLangAcq (Lee et al., 2016).

3.1.2 Standard Hebrew Berman Longitudinal CHILDES Corpus

The Standard Hebrew Berman Longitudinal CHILDES corpus has the same talks as in 3.1.1, but written in standard Hebrew. This corpus has only raw data of Hebrew text, while the original one, transcribed in latin-based phonemic, has also morphological annotations as metadata.
Our objective is to merge these datasets, creating a comprehensive dataset that incorporates Hebrew text along with all the annotations from 3.1.1, both at the utterance level and the word level. The annotations are needed for the creation of the HeCLIMP evaluation benchmark (See Section 4).

3.2 Corpora Merge and Data Preprocessing

The corpora merge involves file-level, utterance-level, and token-level matching. As part of the corpora merge, we performed data cleaning, which included morphological segmentation (More et al., 2019) and punctuation correction (See Appendix A for more details). The resulted corpus **HTBerman** (Hebrew Transcription Berman) contains 53K sentences, 233k words and ∼8K unique words of Hebrew transcribed CDS.

4 HeCLiMP Evaluation Benchmark

We compile HeCLiMP (Hebrew Child-Directed Linguistic Minimal Pairs), a Hebrew CDS grammar test suite, to evaluate how well language models grasp grammaticality in an environment that closely reflects the linguistic input children receive.

Based on minimal pairs (Conneau et al., 2018), HeCLiMP is composed of sentence pairs that differ by just one key element — one sentence is grammatically correct and the other is minimally incorrect. We focus on two grammatical phenomena, adapting Determiner-Noun (DN) agreement from BLiMP and Zorro to Hebrew. By doing so, we address a phenomenon that exists in English (number agreement) and one that does not hold in English (gender agreement):
(1) **DN Number Agreement**: e.g., *'ha-kova ha-ze' ('this hat'-singular) vs. 'ha-kovaim ha-ele' ('these hats'-plural).*
(2) **DN Gender Agreement**: Unlike English, Hebrew requires determiners to match the gender of the noun , e.g., *'ha-kova ha-ze' ('this hat'-masc.) vs. 'ha-simla ha-zo' ('this dress'-fem.).*

Following the procedure used for Zorro in the case of English, we generated minimal pairs using template filled with words from HTBerman (Section 3). Each paradigm consists of 5,596 minimal pairs in the test set and 1,398 minimal pairs in the development set.

Most existing grammar evaluation benchmarks in NLP focus on adult-directed language, posing challenges for assessing the grammatical competence of models trained on CDS. To address this gap in the case of Hebrew, we developed HeCLiMP, a benchmark specifically designed to evaluate Hebrew grammatical learning in models trained on CDS. Our approach follows the methodology of BLiMP and Zorro, but with simplified templates that prioritize morphological features relevant to Hebrew language acquisition.

To construct test sentences, we first designed a set of sentence templates for each grammatical paradigm. These templates were then populated with words sampled from HTBerman, ensuring that all inserted content words conformed to the necessary morphological constraints. Word lists were generated by filtering nouns from HTBerman along with their gender and number annotations.

A primary focus of HeCLiMP is determiner-noun agreement in Hebrew, specifically gender and number agreement. We used simple templates such as *"Look at this ..."* or *"Look at that ..."*, where the determiner adapted according to the gender and number of the noun.

	HeRo	TafBERTa
Parameters	125M	3.3M
Data size	47.5GB	1.8MB
Words in data	4.7B	233k
Batch size	8k	128
Max sequence	512	128
Epochs	25	5
Hardware	1xGTX1080	1xRTX6000
Training time	35 days	105 seconds
Model Configurations		
Vocabulary size	50K	7317
Hidden size	768	64
Layers number	12	10
Attention heads	12	4
Intermediate size	3072	2048
Max. sequence	512	128
Accuracy *	73.5	69.4

Table 1: A Comparison between HeRo, pre-trained on 4.7B words of web text, and TafBERTa, pre-trained from scratch on 233k words of child-directed input. *Accuracy results on the evaluation task.

5 TafBERTa

Model We introduce a scaled-down masked language model based on RoBERTa, with 3.3M parameters, 7317 vocabulary items trained on 233K words. We will refer to this model as TafBERTa[4]. All hyper-parameters were identified by tuning TafBERTa on a masked word prediction task using a held-out portion of our corpus of transcribed CDS as input. A detailed comparison between hyper-parameters of TafBERTa and other Hebrew LMs we compared to is in Tables 1 and 2. Briefly, TafBERTa uses only 10 layers, 4 attention heads, 64 hidden units and an intermediate size of 2048.

Vocabulary TafBERTa uses a Byte-Pair Encoding (Sennrich et al., 2016) sub-word vocabulary, like HeRo and RoBERTa. Instead of HeRo's 50K word vocabulary, we built a 7317-word vocabulary from HTBerman.

Hyper-Parameters Search We optimized hyper-parameters on the development set (Table 5), focusing on those with significant improvements in BabyBERTa. The development set consisted of the two DN agreement paradigms from HeCLiMP.

6 Experiments

Results reflect the average performance over six runs with different seeds including RoBERTa on HTBerman (§6.2), BabyBERTa on HTBerman (§6.2), and the Wikipedia-trained model (§7).

The models used for the comparison are HeRo [5], AlephBERT [6], AlephBERTGimmel [7], DictaBERT [8], and DictaLM2.0 [9]. Differently from the other models, which are encoder-based language models, DictaLM2.0 is a large decoder-based language model. A comparison between the models is presented in Table 2.

6.1 Evaluation Method

Inspired by the BabyBERTa paper, we use *holistic scoring* (Zaczynska et al., 2020). For each minimal pair, we calculate the model's preference for the grammatical sentence over the ungrammatical one. This score is obtained by summing the cross-entropy errors across all positions in the sentence. Accuracy is the ratio of correct choices to total pairs.

6.2 Results

The results are presented in Table 3. These questions (Q1, etc.) are as described in the introduction.

[4]Taf means toddler in Hebrew

[5]https://huggingface.co/HeNLP/HeRo

[6]https://huggingface.co/onlplab/alephbert-base

[7]https://huggingface.co/imvladikon/alephbertgimmel-base-512

[8]https://huggingface.co/dicta-il/dictabert

[9]https://huggingface.co/dicta-il/dictalm2.0

Model	AlephBERT	AlephBERTGimmel	DictaBERT	DictaLM 2.0	HeRo	TafBERTa
Parameters	126M	184M	184M	7B	125M	**3.3M**
Words in data	1.9B	2B	3.8B	35B*	4.7B	**233K**

Table 2: Comparison of model sizes and training data.

Model	Overall	Number	Gender	Tokenizer	Model
AlephBERT	58.6	58.7	58.5	WP	Encoder
AlephBERT-Gimmel	59.8	54.3	65.3		
DictaBERT	87.1	90.1	84.2		
DictaLM2.0	46.1	31.4	60.8	BPE	Decoder
HeRo	73.5	69.1	77.9	BPE	Encoder
RoBERTa (HTBerman)	65.6	83.7	47.5		
TafBERTa	69.4	80.5	58.2		

Table 3: Accuracy on each phenomenon in HeCLiMP. We used the Holistic-scoring method. “Overall” refers to the overall accuracy across all phenomena. “Number” and “Gender” refer to determiner-noun agreement in number and gender, respectively. WP refers to the WordPiece tokenizer.

Model	Overall	Number	Gender
Wikipedia	43.3	30.9	55.7
TafBERTa	**69.4**	**80.5**	**58.2**

Table 4: Performance of the Wikipedia-Trained Model and TafBERTa on the HeCLIMP subset. "Overall" refers to the overall accuracy across all phenomena. “Number” and “Gender” refer to determiner-noun agreement in number and gender, respectively. The highest score in each column is highlighted in bold.

Comparison with HeRo (Q1) Since HeRo and TafBERTa share the same architecture and tokenizer, the comparison between the two allows for a direct assessment of the impact of training data and optimization choices. TafBERTa achieved an overall accuracy of 69.4 on the test set while HeRo reaches 73.5. Breaking this down by task, we observe an interesting tradeoff: while TafBERTa excels in DN agreement for number (80.5 vs. 69.1), HeRo demonstrates superior performance in DN agreement for gender (77.9 vs. 58.2).

Comparison to RoBERTa trained on HTBerman (Q2) We trained the RoBERTa architecture on HTBerman using the same number of epochs as TafBERTa. It achieved 65.6 overall accuracy, with strong performance on number agreement (83.7) but poor results on gender agreement (47.5). This highlights the importance of tailored pre-training objectives and hyperparameter optimization, as seen in TafBERTa, to achieve balanced performance across linguistic tasks. A further analysis of RoBERTa is presented in Appendix D.

Comparison to BabyBERTa TafBERTa and BabyBERTa share the same underlying architecture, but differ in their hyper-parameters. To directly compare the two, we trained BabyBERTa's architecture using its original hyper-parameters on HTBerman. BabyBERTa achieved lower performance than TafBERTa on the two tasks, suggesting that careful adaptation of hyper-parameters is crucial when applying a shared architecture to different languages.

Additional comparisons (Q3) We observe that DictaLM 2.0, a Large Language Model being the current state-of-the-art (SOTA) for Hebrew in general tasks, performed the worst on the number agreement task, achieving only 31.4 accuracy, sig-

nificantly below other models.

In the group of RoBERTa-based models using WordPiece tokenizers, DictaBERT achieved the highest overall accuracy in this group (87.1), with especially strong results in number agreement (90.1). In contrast, AlephBERT and AlephBERT-Gimmel lagged behind, with overall accuracies of 58.6 and 59.8, respectively, reflecting less robust handling of grammatical tasks.

7 Alternative Training Data

We assess the adaptability of TafBERTa's architecture by testing its ability to learn from alternative data sources, specifically evaluating its performance when trained on Wikipedia-derived Hebrew text rather than CDS (**Q4**). We utilized the SVLM Hebrew Wikipedia Corpus[10], preprocessed in the same manner as HTBerman. The dataset size was adjusted to match the word count of HTBerman, ensuring equivalent scales for training.

Using this dataset, we trained a new language model that retained the architecture of TafBERTa but replaced the training data with the processed Wikipedia corpus. Subsequently, we evaluated this new model on a subset of HeCLIMP, focusing on minimal pairs containing words seen by the model during training. For comparison, we also assessed TafBERTa on the same test set. That is to say, the two models we compare have seen during training the words used in the benchmark, and only differ by the type of the training data used (HTBerman vs. Wikipedia).

The results (Table 4) indicate that while the Wikipedia corpus serves as a rich and diverse resource, its effectiveness in training for grammatical agreement tasks is limited compared to the original dataset used for TafBERTa.

8 Conclusion

We present in this paper TafBERTa, a first language model tailored to Hebrew Child-Directed Speech. Focusing on Determiner-Noun agreement phenomena, we show that TafBERTa shows competitive performance with larger Hebrew language models. By doing so, we extend acquisition-inspired, small-scale language model research to a low-resource language, where such efforts are particularly needed. Our results emphasize the need for language-specific and data-specific tuning to fully leverage the capabilities of such models. Future work includes the extension of HeCLiMP to additional grammatical phenomena, the use of training data originated from later stages of language acquisition (i.e., language directed to older children), and the exploration of alternative language model architectures.

[10] https://github.com/NLPH/SVLM-Hebrew-Wikipedia-Corpus

Limitations

While TafBERTa demonstrates progress in modeling Hebrew child-directed speech, several limitations highlight areas for future work and improvement.

Evaluation Improvements Our evaluation framework, HeCLiMP, successfully benchmarks grammatical proficiency but remains limited in scope. Currently, it focuses on determiner-noun agreement in gender and number. Future work should expand HeCLiMP to include a set of grammatical structures, such as verb-subject agreement and determiner-noun agreement with an adjective in between.

Multilingual Model Development While TafBERTa is optimized for Hebrew, its application is restricted to a monolingual context. Extending the model to a multilingual framework by training on related Semitic languages (e.g., Arabic) could enhance its ability to generalize across linguistic variations.

Training on Older Children's Data Currently, TafBERTa is trained on speech data directed at younger children, which captures early-stage language acquisition patterns. However, language complexity increases with age. Training on speech data directed at older children would enable the model to learn more advanced syntactic and morphological structures, better simulating additional phases of language development.

Exploring Alternative Architectures The BERT architecture has dominated Hebrew NLP research and TafBERTa follows this trend. However, exploring other architectures may yield performance improvements. Additionally, architectures optimized for low-resource settings, such as efficient transformers (e.g., DistilBERT (Sanh et al., 2020)), could offer a better trade-off between computational efficiency and linguistic expressiveness.

Ethics Statement

The language acquisition data we are using in this work were taken from the TalkBank system[11], with includes CHILDES, and where all contributions have received an IRB approval. Our own work on the data has been approved by the Ben-Gurion University of the Negev Ethics committee.

Acknowledgements

We would like to thank Shuly Wintner and Bracha Nir for sharing with us the CHILDES data converted into Hebrew script, which forms the basis of the HTBerman corpus presented in this work, and the anonymous reviewers for their helpful comments. We also acknowledge the NICHD HD082736 grant support for CHILDES. Our work was supported in part by grants from the Israeli Ministry of Innovation, Science & Technology (#000519) and from the Data Science Research Center at Ben-Gurion University of the Negev.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631.

Aviad Albert, Brian MacWhinney, Bracha Nir, and Shuly Wintner. 2012. A morphologically annotated hebrew childes corpus. In *Proc. of the Workshop on Comp. Models of Language Acquisition and Loss*, pages 20–22.

Sharon Armon-Lotem. 1996. *The minimalist child: Parameters and functional heads in the acquisition of Hebrew*. Tel Aviv University.

Bastian Bunzeck, Daniel Duran, and Sina Zarrieß. 2025. Do construction distributions shape formal language learning in German BabyLMs? In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 169–186, Vienna, Austria. Association for Computational Linguistics.

Luca Capone, Alice Suozzi, Gianluca Lebani, and Alessandro Lenci. 2024. Babies: A benchmark for the linguistic evaluation of italian baby language models. In *Italian Conf. on Comp. Ling.*

Filippo Chiarello, Vito Giordano, Irene Spada, Simone Barandoni, and Gualtiero Fantoni. 2024. Future applications of generative large language models: A data-driven case study on chatgpt. *Technovation*, 133:103002.

[11] https://childes.talkbank.org/

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Bar Gazit, Shaltiel Shmidman, Avi Shmidman, and Yuval Pinter. 2025. Splintering nonconcatenative languages for better tokenization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22405–22417, Vienna, Austria. Association for Computational Linguistics.

Zebulon Goriely and Paula Buttery. 2025. IPA CHILDES & G2P+: Feature-rich resources for cross-lingual phonology and phonemic language modeling. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 502–521, Vienna, Austria. Association for Computational Linguistics.

Kyle Gorman and Yuval Pinter. 2025. Don't touch my diacritics. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 285–291, Albuquerque, New Mexico. Association for Computational Linguistics.

Eylon Gueta, Avi Shmidman, Shaltiel Shmidman, Cheyn Shmuel Shmidman, Joshua Guedalia, Moshe Koppel, Dan Bareket, Amit Seker, and Reut Tsarfaty. 2023. Large pre-trained models with extra-large vocabularies: A contrastive analysis of hebrew bert models and a new one to outperform them all. ArXiv 2211.15199 [cs.CL].

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proc. of NAACL'18*, pages 1195–1205.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Philip A. Huebner and Jon A. Willits. 2021. Chapter eight - using lexical context to discover the noun category: Younger children have it easier. In Kara D. Federmeier and Lili Sahakyan, editors, *The Context of Cognition: Emerging Perspectives*, volume 75, pages 279–331. Academic Press.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Marvin Lavechin, Yaya Sy, Hadrien Titeux, María Andrea Cruz Blandón, Okko Räsänen, Hervé Bredin, Emmanuel Dupoux, and Alejandrina Cristia. 2023. Babyslm: language-acquisition-friendly benchmark of self-supervised spoken language models. In *INTERSPEECH 2023*. ISCA.

Jackson L. Lee, Ross Burkholder, Gallagher B. Flinn, and Emily R. Coppess. 2016. Working with chat transcripts in python. Technical report, Department of Computer Science, University of Chicago.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for analyzing talk, Third Edition*. Mahwah, NJ: Lawrence Erlbaum Associates.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*

Amir More, Amit Seker, Victoria Basmova, and Reut Tsarfaty. 2019. Joint transition-based models for morpho-syntactic parsing: Parsing strategies for MRLs and a case study from Modern Hebrew. *Transactions of the Association for Computational Linguistics*, 7:33–48.

Aaron Mueller and Tal Linzen. 2023. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11237–11252, Toronto, Canada. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Miyu Oba, Tatsuki Kuribayashi, Hiroki Ouchi, and Taro Watanabe. 2023. Second language acquisition of neural language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13557–13572, Toronto, Canada. Association for Computational Linguistics.

Riccardo Orlando, Luca Moroni, Pere-Lluís Huguet Cabot, Simone Conia, Edoardo Barba, Sergio Orlandini, Giuseppe Fiameni, and Roberto Navigli. 2024. Minerva llms: The first family of large language models trained from scratch on italian data. In *Italian Conf. on Comp. Ling.*

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is more: Pre-training cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 174–188, Miami, FL, USA. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. AlephBERT: Language model pre-training and evaluation from sub-word to sentence level. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–56, Dublin, Ireland. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Vitaly Shalumov and Harel Haskey. 2023. Hero: Roberta and longformer hebrew language models. *arXiv preprint arXiv:2304.11077*.

Zhewen Shen, Aditya Joshi, and Ruey-Cheng Chen. 2024. BAMBINO-LM: (bilingual-)human-inspired continual pre-training of BabyLM. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–7, Bangkok, Thailand. Association for Computational Linguistics.

Shaltiel Shmidman, Avi Shmidman, Amir DN Cohen, and Moshe Koppel. 2024. Adapting llms to hebrew: Unveiling dictalm 2.0 with enhanced vocabulary and instruction capabilities. *Preprint*, arXiv:2407.07080.

Shaltiel Shmidman, Avi Shmidman, and Moshe Koppel. 2023. Dictabert: A state-of-the-art bert suite for modern hebrew. ArXiv 2308.16687 [cs.CL].

Taiga Someya and Yohei Oseki. 2023. JBLiMP: Japanese benchmark of linguistic minimal pairs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1581–1594, Dubrovnik, Croatia. Association for Computational Linguistics.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece tokenization. In *Proc. of EMNLP'21*, pages 2089–2103.

Reut Tsarfaty, Shoval Sadde, Stav Klein, and Amit Seker. 2019. What's wrong with Hebrew NLP? and how to make it right. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International*

Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, pages 259–264, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *TACL*, 8:377–392.

Beilei Xiang, Changbing Yang, Yu Li, Alex Warstadt, and Katharina Kann. 2021. CLiMP: A benchmark for Chinese language model evaluation. In *Proc. of EACL*, pages 2784–2790.

Aditya Yadavalli, Alekhya Yadavalli, and Vera Tobin. 2023. SLABERT talk pretty one day: Modeling second language acquisition with BERT. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11763–11777, Toronto, Canada. Association for Computational Linguistics.

Karolina Zaczynska, Nils Feldhus, Robert Schwarzenberg, Aleksandra Gabryszak, and Sebastian Möller. 2020. Evaluating German transformer language models with syntactic agreement tests. In *Proceedings of the 5th Swiss Text Analytics Conference and the 16th Conference on Natural Language Processing, SwissText/KONVENS 2020, Zurich, Switzerland, June 23-25, 2020*, volume abs/2007.03765, Zurich, Switzerland. CEUR Workshop Proceedings.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. A survey of large language models. ArXiv 2303.18223 [cs.CL].

A Data Preprocessing for the HTBerman Construction

Our ultimate aim is to train TafBERTa using Hebrew Child-Directed Speech data. To accomplish this, we must filter the CDS utterances in Standard Hebrew corpus 3.1.2, while the label of the speaker appears in 3.1.1. The primary task during the preprocessing phase involves merging the corpora outlined at 3.1.

A.1 File-level matching

In Hebrew corpus files, there is incompatibility in files' order with English Berman longitudinal dataset. In order to overcome this problem, we made manual changes to the Hebrew corpus, including removing blank files and reordering according to Berman longitudinal dataset files' order.

A.2 Utterance-level matching

In the datasets, most files contain an equal number of lines, except for certain files within the English Berman longitudinal dataset. These additional lines are filled with irrelevant or duplicate information compared to the standard Hebrew data. We manually identified and removed these lines from the English dataset. In this corpus, there are 268 files, out of them 64 are found to be problematic.

A.3 Token-level matching

Matching tokens for each pair of English-Hebrew sentences often leads to numerous conflicts within the sentence (in token level). There are several types of gaps that lead to these conflicts. In the process of overcoming the gaps, we edit the Hebrew sentences in an automatic script.
Here are our primary steps to align as many sentences as feasible, focusing solely on editing Hebrew sentences:

- Create segmented sentences using YAP (Yet Another (natural language) Parser)(More et al., 2019)[12].
- Merge children's names (Hagar, Leor, Lior) to single names instead of separated (for example, Ha gar to Hagar).
- Combine separated words that should be one word.
- Separate conjunction.
- Remove random "junk" letters in the middle of the sentence.
- Insert spaces between punctuation marks that are directly attached to text.
- If punctuation is absent in a Hebrew sentence as it appears in the Latin transcription, add the appropriate punctuation marks.

[12]Apache-2.0 license

- Correct prepositions (if they are written separated in Latin transcription but connected to words in Hebrew).
- Correct double words (combine words like "Od Paam" to "Od_Paam", as in English it appear as a single word - again). We carried out this step at this point rather than earlier because in the preceding sections, we addressed all aspects concerning word indexes when sentences are segmented by spaces.
- Attempt to correct the prepositions once more, considering that the indexes may have changed after addressing duplicate words.

Please note that during the correction of prepositions, we proceeded to the next step only if our function successfully rectified the sentence. If the correction was not made, the incorrect sentence was retained for another attempt.

B Implementation Details and Reproducibility

All experiments were run for 100 epochs, with each run taking approximately 15 minutes of training. For each run, we identified the epoch at which the maximum accuracy on the development set was achieved (referred to as the "max epoch"). The final reported result for each run is the test accuracy at this "max epoch".

During the process, we logged two models in MLflow for each run: the model corresponding to the "max epoch" and the model after completing all 100 epochs. As the top-performing runs showed minimal variation in development and test accuracy, we further refined the process by training the model for each hyper-parameter combination using six different random seeds. The final selected model for each configuration was the one with the highest average development accuracy across these seeds.

B.1 Hyper-parameter optimization

Hyper-parameter optimization was conducted using Optuna (Akiba et al., 2019)[13], an open-source framework designed for efficient and automated hyper-parameter tuning. Optuna employs techniques such as Bayesian optimization and pruning mechanism to enhance search efficiency and reduce computational costs. The optimization process was guided by a defined objective function; maximize the accuracy on the development set and evaluating performance metrics on accuracy and loss.

All experimental results, including hyper-parameter trials, best-performing configurations and model performance metrics, were systematically logged using MLflow[14]. MLflow provided experiment tracking, reproducibility and model versioning, enabling comprehensive monitoring and comparison of different hyper-parameter tuning runs.

B.2 Model Logging

Both the final and best performing models were logged using MLflow. For each of the runs, the model on the last epoch and the best model of the run, selected based on *accuracy_dev_max*, is available for future benchmarking.

C RoBERTa Optimized

In addition to the use of the RoBERTa architecture with the same number of epochs as TafBERTa (see Section 6, we also explore the optimization of the RoBERTa model given the HTBerman data, increasing the number of epochs. The results are presented in Table 6.

D Results Visualization

This appendix provides the detailed evaluation results of various Hebrew language models on grammatical agreement tasks. The models were assessed on Number Agreement, Gender Agreement and Overall Accuracy using the HeCLiMP benchmark. The figures illustrate the performance of each model with respect to the number of parameters and words seen in the training phase.

D.1 Number Agreement Accuracy

The first evaluation metric focuses on the ability of models to correctly predict number agreement in Hebrew. As shown in Figure 2, DictaBERT achieved the highest accuracy at 90.1%, followed by TafBERTa with 80.5%. HeRo performed at 69.1%, while AlephBERT and AlephBERTGimmel recorded 58.7% and 54.3%, respectively. DictaLM 2.0 performed considerably worse than other models with only 31.4%.

[13] MIT License

[14] https://mlflow.org/ with Apache-2.0 license

Hyper-parameter	Checked Intervals
num_attention_heads	{2, 4, 6, 8, 10, 12}
hidden_size	{64, 128, 256, 512, 768}
leave_unmasked_prob	{0.0, 0.1}
num_layers	{2, 4, 6, 8, 10, 12}
intermediate_size	{64, 128, 256, 512, 1024, 2048, 3072, 4096}

Table 5: Intervals checked for each hyperparameter in the Optuna objective function. The upper bound of the search space corresponds to the hyperparameters of RoBERTa. Thus, TafBERTa's smaller size was not intentionally designed to be compact, but rather emerged as the optimal configuration through hyperparameter tuning.

Model	#epoch	Overall	Number	Gender
RoBERTa (HTBerman)	5	65.6	83.7	47.5
RoBERTa (HTBerman)	43	71.1	83.2	59
TafBERTa	5	69.4	80.5	58.2

Table 6: Accuracy on each phenomenon in HeCLiMP using the Holistic-scoring method. "Overall" refers to the overall accuracy across all phenomena. "Number" and "Gender" refer to determiner-noun agreement in number and gender, respectively. RoBERTa was trained for five epochs, matching TafBERTa's training regime and also for a longer period until convergence. When trained for five epochs, RoBERTa achieved lower overall accuracy (**65.6**) compared to TafBERTa (**69.4**), with higher performance on number agreement (**83.7** vs. **80.5**) but weaker results on gender agreement (**47.5** vs. **58.2**). Training RoBERTa for more epochs improved its overall accuracy (**71.1**) and performance on the gender agreement task (**59**) but slightly reduced its accuracy on number agreement (**83.2**). TafBERTa maintains a better balance across both tasks.

D.2 Gender Agreement Accuracy

Figure 3 demonstrates that DictaBERT again performed the best, reaching 84.2 accuracy. HeRo followed with 77.9, while AlephBERTGimmel and AlephBERT obtained 65.3 and 58.5, respectively. TafBERTa recorded 58.2 and DictaLM 2.0 managed 60.8.

D.3 Overall Accuracy

The overall accuracy metric evaluates the general grammatical understanding of Hebrew language models across different agreement phenomena. Figure 4 shows that DictaBERT leads with an 87.1 accuracy, followed by HeRo at 73.5 and TafBERTa at 69.4. AlephBERT and AlephBERTGimmel achieved 58.6 and 59.8, respectively. DictaLM 2.0 recorded an overall accuracy of 46.1, which is notably lower than the other models.

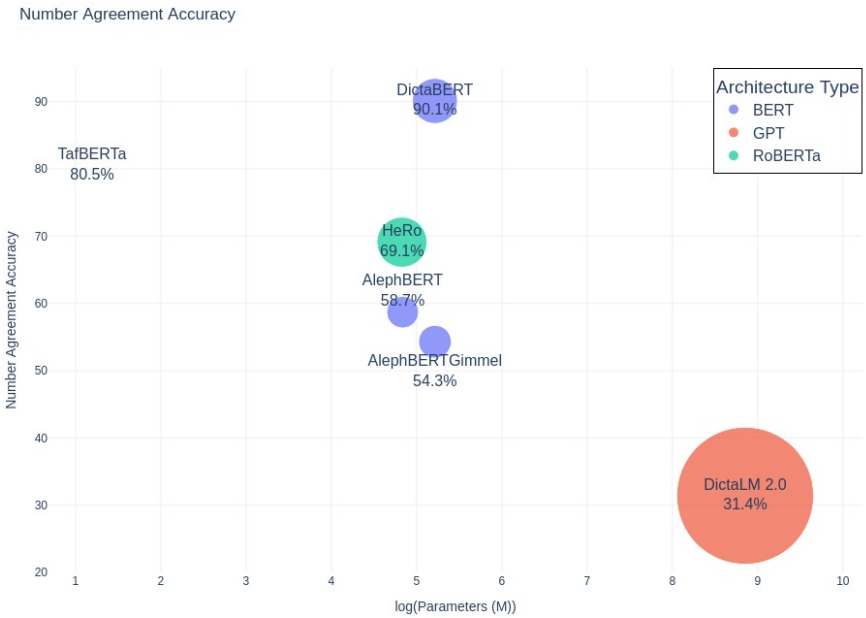


Figure 2: Number Agreement Accuracy of Hebrew Language Models

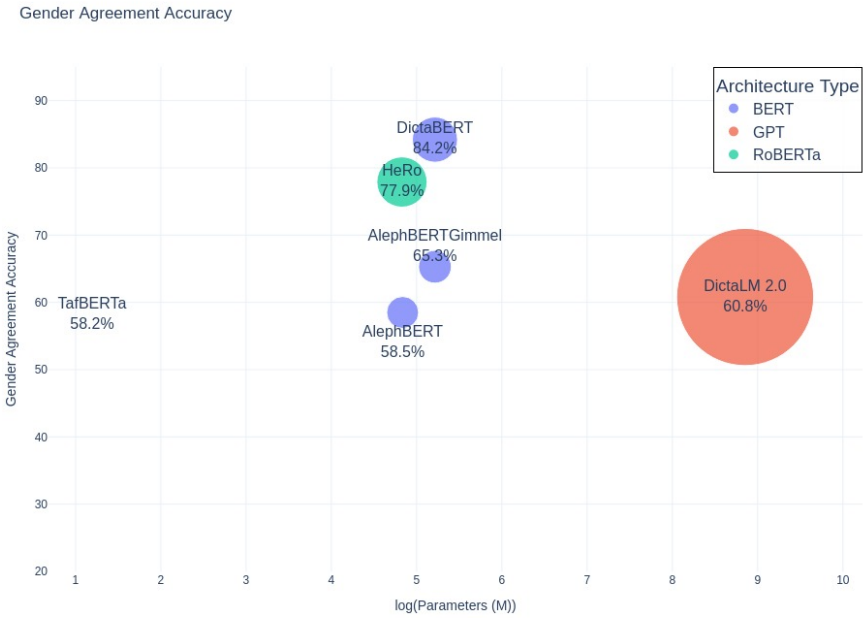


Figure 3: Gender Agreement Accuracy of Hebrew Language Models

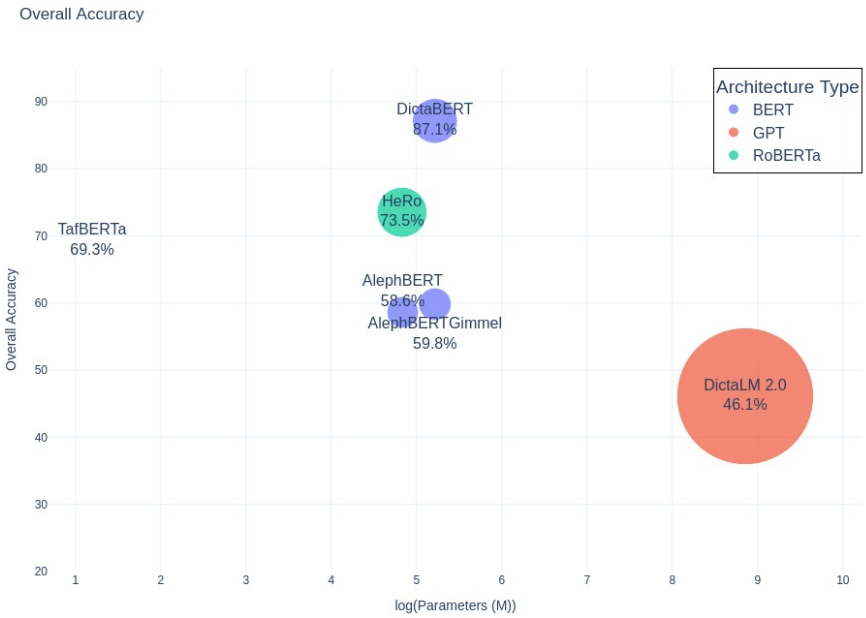


Figure 4: Overall Accuracy of Hebrew Language Models

FORGETTER with forgetful hyperparameters and recurring sleeps can continue to learn beyond normal overtfitting limits

Rui Yamamoto and **Keiji Miura**
Kwansei Gakuin University
1 Gakuen Uegahara, Sanda, Hyogo 669-1330, JAPAN
miura@kwansei.ac.jp

Abstract

LLMs suffer from considerable computational costs in training. A more biologically plausible curriculum learning may help to decrease the learning costs. Here we propose a FORGETTER training algorithm, in which a model forgets the variables for optimization after a sleep and the hyperparameters are set toward forgetting memory: rather large weight decay and learning rates as well as small but optimized batch sizes. By limiting minGemma model to 512 input length and speeding up the development cycle, we compared normal and FORGETTER learning algorithms by using more than a thousand different models. Specifically, we found and utilized the "120-rule" that the models with about 120 (Query) heads in total, irrespective of the head number per layer, outperform. The improvement by using the FORGETTER algorithm is far bigger than that by optimizing the model structure. Specifically, FORGETTER models can learn beyond the data size where the normal learning overfits. The FORGETTER also works for CIFAR10 image classification. These results suggest that forgetting can be beneficial for pretraining deep neural networks by avoiding overfitting.

1 Introduction

Although ChatGPT's performance was amazing enough to revolutionize what are human jobs (OpenAI, 2023; Rothman, 2024), they require considerable computational resources (Tunstall et al., 2022). While many approaches for making the training more efficient have been proposed (Goodfellow et al., 2016; Atienza, 2020; Chollet, 2021; Geron, 2022), recent LLMs are too huge to explore their hyperparameters and learning algorithms exhaustively.

As human babies do not need as many resources to learn a language apparently, a more efficient learning method may remain to be discovered (Ford, 2018; Warstadt et al., 2023). It is true that nowadays a handy Trainer class in PyTorch provides a normal training routine, which enables us to explore various model structures and other hyperparameters quite easily. However, it may be also promising to explore unconventional learning procedures (Smith, 2017; Zhao et al., 2024), possibly learned from the biological brain.

In the same vein, BabyLM data is very suitable to mimic how human babies earn language abilities including grammars (Warstadt et al., 2023; Mahowald et al., 2024) with rather small language models (Raschka, 2024; Lu et al., 2025; Tunador). Small language models allow us to explore more hyperparameters related to learning (Warstadt et al., 2023). Although many models succeeded to learn BabyLM10M/100M in the past contests, the biological plausibility of the training algorithm (Konishi et al., 2023; Lillicrap et al., 2020) was not necessarily pursued.

We believe that two ingredients are important for biological plausibility. First of all, animals are not perfect and inevitable to forget. For example, animals cannot keep huge training data and memory traces without fail. Thus, a biologically plausible model should allow animals to forget to reasonable extent. Furthermore, animals sleep, which also forces them to forget. If these inevitable forgetting is beneficial or not remains an open question. The observations on the effect of sleep on learning in neuroscience (Norimoto et al., 2018) should be incorporated for training LLMs. (The algorithm that works for big data with LLMs is also promising as the computational model of the brain.)

Here we proposed the FORGETTER model, that is trained with sleep and forgetful hyperparameters. We demonstrate that it can learn beyond the data size where the normal learning overfits.

In Section 2, as a methodology, we explain the base model we used and the novel FORGETTER training method, in which we insert sleep (variables for optimization are initialized) between epochs.

Proceedings of the First BabyLM Workshop, pages 91–99

This may be regarded as a procedural learning. In Section 3, as results for Baby10M, we show our FORGETTER model outperforms the normally trained models. Specifically, you can continue and repeat epochs for FORGETTER models, where couples of epochs suffice to excel the normal training. In Section 4, as results for Baby100M, we demonstrate that the benefit of FORGETTER is inherited to Baby100M. It can again learn beyond the normal limits. In Section 5, we demonstrates that the FORGETTER also works for CIFAR10 image classification. In Section 6, as Summary and Discussion, we summarized the results and discussed the strengths of the proposed methods. In Limitation, complement to the discussion of strengths in the main text, we discuss the limit of the proposed methods. Specifically if a FORGETTER model is always better than a normal model is an important question. We discuss how general can the benefit of the FORGETTER be.

2 Methods

All the computation was done by the custom-written Python codes on 7 PCs with NVIDIA RTX 3090, 4080, 4090 or A6000 GPU. All the codes to reproduce this paper's results and the list of validated loss for varieties of model structures trained with normal or FORGETTER algorithms are available at GitHub (https://github.com/keiji-miura/FORGETTER-BabyLM).

2.1 DATA

Baby10M and Baby100M dataset were used for (pre)training of next token prediction. The both Baby10M and Baby100M data were tokenized by the GPT2 Tokenizer. We saved the tokenized data to a single file (separately for Baby10M/Baby100M or training/validation) to speed up I/O during training, in which consecutive 512 tokens were cut out at a random starting point for training data. There, we used different random starting points for different 512 tokens within a batch.

2.2 BASE MODELS

The minGemma model (Tunador; Gemma Team, 2024) was entirely used in this paper as a text generation model. We trained the minGemma model from scratch by using either the BabyLM10M or the BabyLM100M. In this paper, we solely compared the normal and FORGETTER models. The difference between the normal models and FORGETTER models are in the training algorithms.

Hyperparam.	**Baby10M**	**Baby100M**
Tokenizer	GPT2	GPT2
Input Size	512	512
Drop Out	No (p=0)	No (p=0)
Weight Decay	1.0	0.25
Batch Size	12	28
Learning Rate:		
- Normal	1.35×10^{-3}	10^{-3}
- FORGETTER	10^{-3}	0.8×10^{-3}
N Steps/Epoch:		
- Normal	19600	144000
- FORGETTER	10000	70000

Table 1: Hyperparameters. Fixed setting (GPT2 Tokenizer, input length=512, no drop-out) speeded up the development cycle, which enabled us to explore different training methods and varieties of model structures.

2.2.1 model representation

We represent a model by the combination of the numbers such as "L24-6(3)-648×4-240". (These numbers are what we occasionally changes to explore better results.) The meaning of the numbers are the number of layers, the number of Query heads, the number of Key/Value heads (this must divide the number of Query heads), the hidden dimension at the input layer of the feed forward layer, the hidden dimension at the hidden layer of the feed forward layer, the head dimension at attention. Note that, as our minGemma model was Gemma-based (Gemma Team, 2024), we not only explored the number of Query heads but also the number of Key/Value heads in a grouped attention.

2.3 TRAINING ALGORITHM

2.3.1 Normal training algorithm

The normal model was trained by using the PyTorch Trainer class with a single epoch where the learning rate linearly decays to zero. The number of steps in an epoch was optimized so that more steps caused overtraining.

2.3.2 FORGETTER with sleep 1 (light sleep)

The FORGETTER models was trained by repeating the Pytorch Trainer with multiple epochs. There, in each epoch, the learning rate linearly decays to zero. Between epochs, the model was not initialized (specifically the weights were kept) while the variables for AdamW were initialized. That

is, only the optimizer was reinitialized between epochs. This is why we say FORGETTER models "sleep", after which the optimizer is initialized. When we simply mention "sleep", we mean this sleep 1.

2.3.3 FORGETTER with sleep 2 (deep sleep)

Because sleep is a rather vague concept, we can consider another definition for sleep. For "sleep2", not only the optimizers but also all the state variables in the model except weights are initialized. We implemented this simply by loading the pre-dumped weights to a newly constructed minGemma model in PyTorch.

In principle, at the transition of epochs, you can use either sleep 1 or sleep 2. However, we found sleep 2 can be most effective at last. That is, after the repetition of sleep 1, in the final epoch sleep 2 can drop the validated loss largely. We sometimes call this phenomenon "last big drop" by sleep 2.

In this paper, specifically, once (repeated) Sleep 1 overfits (=validated loss increases), it is switched to Sleep 2 (with the model in the previous epoch recovered), although Sleep 2 also overfits soon (at the second time or so) typically.

2.4 FIXED HYPERPARAMETERS for FORGETTING

Here we briefly describe three hyperparameters that are set toward forgetting memory: rather large weight decay and learning rates as well as small but optimized batch sizes.

2.4.1 Weight decay is as large as 1 or 1/4 for Baby10M or Baby100M

Conventionally, a small value of weight decay like as small as 0.01 has been used (see PyTorch document for example). However, we found that a rather large value of the weight decay was beneficial irrespective of the model structures and other hyperparameters. While the optimal weight decay strongly depends on the training data size, it does not strongly depend on the other hyperparameters like model structures, apparently. Therefore, we set weight decay to 1.0 for Baby10M and 0.25 for Baby100M.

Weight decay is the speed to forget weights. So it is convenient from the viewpoint of biological plausibility that the weight decay as large as 1.0 or 0.25 is optimal for pretraining. It seems that animals or babies can forget rather a lot and still achieve the best learning performance, fortunately. (Note that optimality here is regarding the next token prediction.) So we consider that the weight decay value we use throughout the paper is consistent with the idea of forgetful learning.

2.4.2 Batch size is as small as 12 or 28 for Baby10M or Baby100M

The batch size was fixed to 12 for Baby10M or 28 for Baby100M. This is because we believe that, while the optimal batch size strongly depends on the training data size, it does not strongly depend on other the hyperparameters like model structures.

Although 32 can work as well for Baby100M, you need more VRAM in that case. So we chose 28. But it is actually hard to judge which one is better under the high trial-to-trial variability in validated losses.

Batch sizes can be regarded as a memory for recently encountered data. 12 or 28 is rather small and not like 512 or 1024, which are typically used numbers in deep learning. So it is convenient from the viewpoint of biological plausibility that the batch size as small as 12 or 28 is optimal for pretraining. It seems that animals or babies need to memorize only small number of data to achieve the best performance, fortunately. So we consider the batch size is again consistent with the idea of forgetful (=small memory) learning.

2.4.3 Learning rates is as large as 0.001

Within each epoch the learning rate linearly decays to zero. The initial (maximum) learning rate in an epoch was fixed to about 0.001. The value we used may be rather large, compared with the conventinal one like 0.0001 or smaller (see PyTorch Document, for example). However, we found that a rather large value of the learning rate is beneficial irrespective of the model structures and other hyperparameters. That is, we believe that, while the optimal learning rate strongly depends on the training data size, it does not strongly depend on other hyperparameters like model structures. Therefore we set learning rate to 10^{-3}. (To be precise, we used the range from 0.8×10^{-3} to 1.35×10^{-3} as in Table 1.)

A learning rate can be regarded as a rate to forget past encounters. 0.001 is rather big. So it is convenient from the viewpoint of biological plausibility that the learning rate as large as 0.001 is optimal for pretraining. Therefore, animals or babies can forget rather a lot and achieve the best performance, fortunately. So we consider that the learning rate value we use is again consistent with the idea of

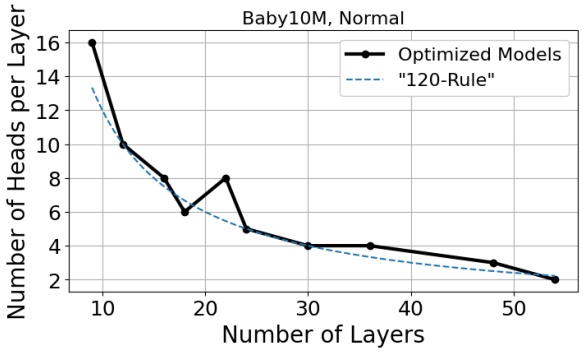


Figure 1: Number of (Query) heads per layer for best models for a given number of layers. These optimal models obey "120-rule" and have about 120 heads in total, irrespectively of the number of layers.

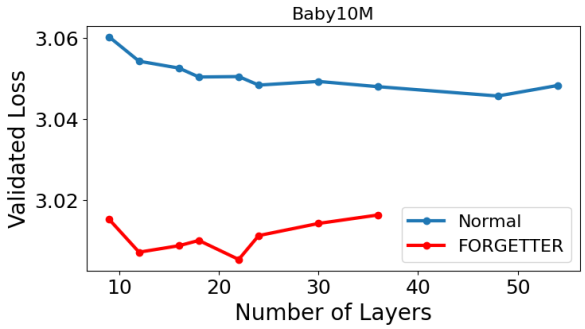


Figure 2: Validated losses for best models for a given number of layers trained with normal or FORGETTER algorithm for Baby10M.

forgetful (=small memory) learning.

3 Results for Baby10M

3.1 120 rule for searching model structure

First, good model structure was searched with the normal training algorithm for Baby10M. The model structures were explored in order to obtain as lowest validated loss as possible. (In this paper, the optimality is always about the validated loss for the next token prediction.)

To see the impact of the number of layers, we searched optimal model structures for a given number of layers (Figure 1). That is, we plotted the number of (Query) heads for the models whose structures are optimized for a given number of layers (by simple grid search for head number, hidden dim etc). Figure 1 demonstrated that, Surprisingly, the optimal models always had about 120 (Query) heads in total irrespective of the number of layers.

Although the result is variable even for the same hyperparameters and model structures, we tried more than a hundred models per layer for BabyLM10M. Therefore we believe the rule is true as an overall tendency. For example, regarding the validated loss for the normal training, it is rather easy to obtain <3.05 for 48-layers models but not for 18-layers models.

The blue line in Figure 2 denotes the validated losses for the same models as in Figure 1 for normal training. The deeper models performed well in general and the 48-layers model showed the best performance (3.0457).

54-layers models were worse, possibly because they have limited options on head numbers per layer. For example, 3 heads per layer is too much (3 x 54 = 162 heads in total), but 2 heads per layer is too little (2 x 54 = 108 heads in total).

3.2 NOTE: model search range can be limited

We succeeded not only to fix considerable hyperparameters without losing performance as in Table1, but also to limit the model structure search range.

The head dimension, the dimension projected immediately before the attention, was only explored from 96 to 352, because optimal values for a given number of layers were always between 192 to 288 for Baby10M trained with normal algorithms (i.e., in the well-explored category).

As the optiml number of Query heads per model was always around 120, we only needed to try limited ranges. For example, when we explored twelve-layer models, the models with ten or twelve Query heads tended to perform very well while too many or too little heads did not perform well. Sometimes we call this observational fact "120-rule" for short.

The hidden dimension for the token representation tended to be optimized around 700. So we only chose some value close to that within the multiples of the number of Query heads. For Baby10M trained with normal algorithms (i.e., a well-explored category), the optimal models for a given number of layers had from 576 to 832 dimensions to represent a single token.

The dimensions of the hidden layer of the feed forward layer is always fixed to the four times that of the input layer of the same feed forward block. Although we have changed from $\times 4$ to $\times 3$, $\times 5$, $\times 6$, $\times 8$, we could not observe significant improvements. (Consider this "$\times 4$" as a fixed parameter.)

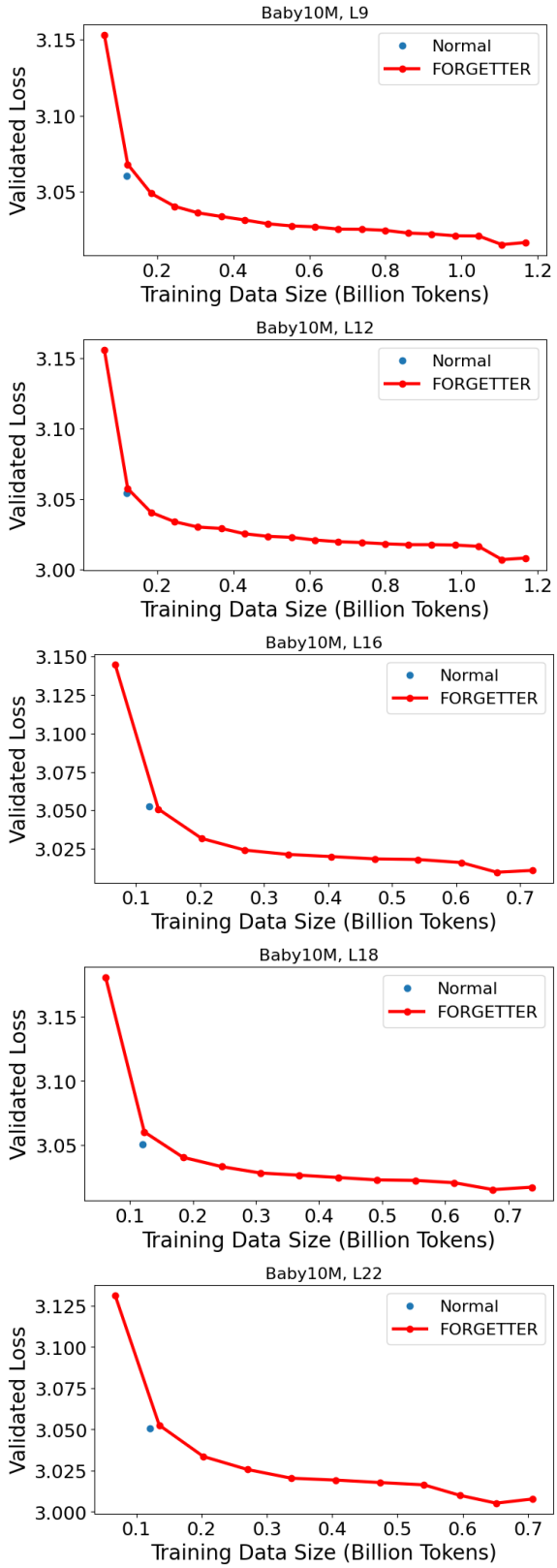


Figure 3: Five examples of training course of FORGETTER algorithm for Baby10M for models with 9, 12, 16, 18 and 22 layers. Validated losses for FORGETTERs are plotted as time series in red. The validated losses for normal models are denoted by a blue point.

Hyperparameter	Normal	FORGETTER
N Layers	48	22
N Q Heads/Layer	3	8
N K/V Heads/Layer	1	4
Hidden Dimension	648	672
FFI Dimension	648×4	672×4
Head Dimension	288	192
Validated Loss	3.0457	3.0053
BLiMP Score	0.6958	0.7257

Table 2: Optimal model structures for Baby10M when input token length is 512.

3.3 Normal vs FORGETTER

The results in Figure 2 demonstrate that FORGETTER models are always much better than normal models. Although the 48-layers model was the best for the normal training, the 22-layers model turned out to be the best for the FORGETTER training.

Note that the difference between normal and FORGETTER models are much larger than that by model structures. This means that the model structure search is not that fruitful. Rather, changing the learning curriculum to FORGETTER is much more efficient way to improve the performance.

In fact, if you look at the time course of the training, the FORGETTER model excels the normal model within a couple of iterations as shown in Figure3. The sleep interval of FORGETTER, that is optimized to minimize the validated loss for next token prediction task, was about half of that of normal models. This is shown as the number of steps per epoch in Table 1. (In normal models, there is no sleep and the entire training consists of only a single interval or epoch.) Therefore the computational time for the two epochs for FORGETTER models is roughly equivalent to that for normal models. At that time, their performances are almost equal. However, FORGETTER models can continue to learn beyond the normal limit as in Figure3. Note that the number of steps per epoch for the normal model (=19600) was already optimized. That means if you used longer steps (more training data) per epoch, the model would overfit and its validated loss deteriorates. Thus, it is interesting that FORGETTER can continue to learn beyond the normal overfitting limits.

Surprisingly, there is a drop in the end of the training (Figure 3). This was caused by sleep 2 (deep sleep). This drop is commonly observed among the models with different layers. The im-

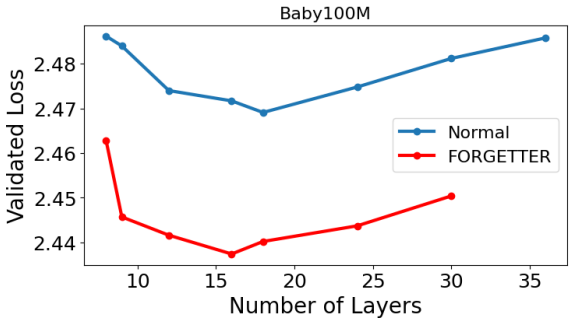


Figure 4: Validated losses for best models for a given number of layers trained with normal or FORGETTER algorithm for Baby100M.

pact of the last big drop is much larger than that by the difference of the model structures. Again, this means that the model structure search is not that fruitful. Rather, changing the learning curriculum to the FORGETTER (with sleep 2 in the end) is much more efficient way to improve the performance.

Remember that our FORGETTER algorithm is not just a learning rate scheduler, but, it randomly initializes state variables after each sleep. The last drop demonstrates how the initialization after the sleep is effective. As the effect of last drop (sleep 2 in the end) is rather variable, the resulting validated loss for the FORGETTER fluctuates across layers (the red line in Figure 2).

Table 2 summarized the best normal and FORGETTER models for Baby10M, where we also computed the BLiMP Score. (We believe that BLiMP Score is almost in one-to-one correspondence to the validated loss for the next token prediction. This is because we have not observed the contradictory results before.) The BLiMP score is (inversely) related to the validated loss for the next token prediction in the current case. The absolute value of BLiMP Score is rather limited because the models were trained with only Baby10M dataset.

4 Results for Baby100M

We trained the minGemma models with variable structures with normal or FORGETTER training algorithm for Baby100M, specifically, to see if the FORGETTER is also effective for Baby100M. The results in Figure 4 demonstrate that the FORGETTER models are always much better than the normal models, again.

To see if the number of layers matters, we plotted the validated losses for our best models for a

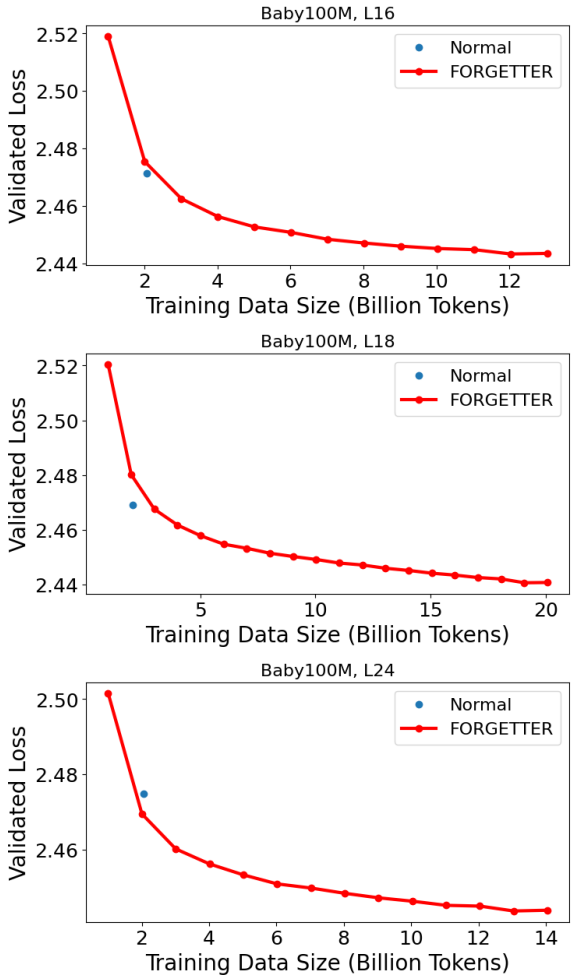


Figure 5: Three examples of training course of FORGETTER algorithm for Baby100M for models with 16 (top), 18 (middle), and 24 layers (bottom). Validated losses for FORGETTERs are plotted as time series in red. The validated losses for normal models are denoted by a blue point.

given number of layers in Figure 4. We observed a clear trough in the plot and the model with 18 layers is the best for both normal and FORGETTER training for Baby100M. (However, we should admit that our structure search may not be complete for Baby100M, whose computational time is rather long.)

Note that the difference between normal and FORGETTER models are much larger than that by model structures. This means that changing the learning curriculum to FORGETTER was the most efficient way to improve the performance.

Next, we looked at the time course during training (Figure 5). Again, the FORGETTER model excels the normal model within a couple of iterations as shown in Figure 5. The sleep interval for the FORGETTER, that is optimized to minimize

Hyperparameter	Normal	FORGETTER
N Layers	18	16
N Q Heads/Layer	8	9
N K/V Heads/Layer	4	3
Hidden Dimension	576	612
FFI Dimension	576×4	612×4
Head Dimension	256	224
Validated Loss	2.4691	2.4374
BLiMP Score	0.7669	0.7761

Table 3: Optimal model structures for Baby100M when input token length is 512.

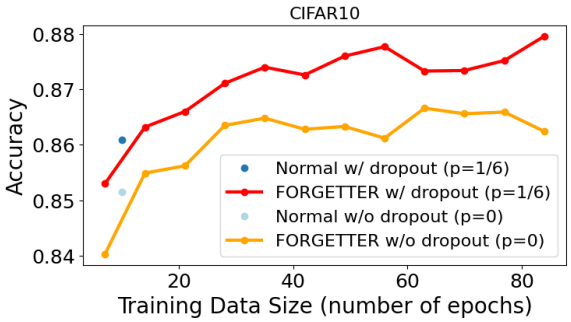


Figure 6: Accuracy of CIFAR10 classification for normal or FORGETTER algorithm with and without dropout. $p = 0$ represents the case without dropout.

the validated loss for next token prediction task, was about half of that of normal models. This is shown as the number of steps per epoch in Talbe 1. (In normal models, there is no sleep and the entire training consists of only a single interval or epoch.)

Therefore the computational time for the two epochs for FORGETTER models is roughly equivalent to that for normal models. At that time, their performances are almost equal. However, FORGETTER models can continue to learn beyond the training data size where normal models overfit as in Figure5. Note that the number of steps per epoch for the normal model (=144000) is already optimized, which means that if you used longer steps (more training data) per epoch, the model would overfit and its validated loss deteriorates.

The drop in the end of the training caused by sleep 2 (deep sleep) was also effective but mild for Baby100M (Figure 5).

Table 3 summarizes the best normal and FORGETTER models for Baby100M, where we also computed the BLiMP Score. Note that the validated losses for Baby100M is much smaller than that of Baby10M. The BLiMP score is in one-to-one correspondence to the validated loss for the next token prediction in the current case.

The absolute value of our BLiMP Score is rather mild (cf. 47.7, 46.2, 78.2 and 79.1 for GPT2 Small, Medium, Large and XL, respectively.) This is partly because the input token length was limited to 512 entirely in this paper. Having shorter input token lengths is as if setting another task. It can impose the upper limit for performances. Although we believe that the comparison between normal and FORGETTER models gave a general result, trying longer input token lengths toward contest quality will be needed in the future work.

5 Results for CIFAR10

We examined if FORGETTER is also effective for CIFAR10 image classification by CNNs. The result in Figure 6 demonstrates that the FORGETTER with a CNN used in a tutorial (Sayah, 2022) can again continue to learn beyond the normal overfitting limit with or without dropout.

The optimal training data size for the normal training with linear learning rate decay was 10 epochs (=10 repeats of the entire training dataset) and longer training caused overfitting. Meanwhile, the optimal inter-sleep interval for the FORGETTER was 7 epochs. That is, the learning rate linearly decayed within every 7 epochs there (the learning rate is reset after 7 epochs). Thus, it is natural that the accuracy of the FORGETTER after one sleep or at 7 epochs is worse than that of the normal model, that learns for 10 epochs. But it exceeds after two sleeps or at 14 epochs. Then, it continues to avoid overfitting for a long time.

These observations are common with or without dropout. Although dropouts might not be strictly needed for pretraining, the impact of dropout is interesting in the sense it can somehow have a similar effect as forgetting. Forgetful hyperparameters and sleeps that initialize optimizers might enhance the redundancy and robustness of representations, which can be a similar role as dropout.

In fact, the dropout ($p = 1/6$) is not only effective but also synergistic with the FORGETTER curriculum learning, suggesting that the mechanism of FORGETTER may be independent from that of dropout. Specifically, the FORGETTER with dropout ($p = 1/6$) attained 88% (87.96%) even without data augmentation. Note that the same code with normal training attains 88% only with data augmentation (Sayah, 2022).

6 Summary and Discussion

We extensively explored the optimal model structure for Baby10M and found "120-rule" where optimal models always almost have 120 (Query) heads. This suggests that there is a specific number (=120) of information processing the model has to treat. And the optimal model tends to have this number of attention heads. We compared normal and FORGETTER models for Baby10M and found that FORGETTER models performed much better. This tendency also holds for Baby100M, in which the best performance was much better. The FORGETTER also worked for CIFAR10 image classification. Overall, FORGETTER models can continue to learn beyond the normal overfitting limits. These results suggest that forgetting can be beneficial for pretraining deep neural networks by avoiding overfitting.

The FORGETTER training can bring about Copernican Revolution on overfitting. It is beneficial if you can control to train beyond the normal overfitting limits.

Regular sleep intervals (number of steps per epoch) apparently worked, once interval lengths were carefully optimized as in Table 1. We could not get a significantly better result by using linearly increasing/decreasing sleep intervals. Also we could not get a significantly better result by using linear increasing/decreasing initial learning rates across epochs.

120 rule saves your computational cost for model structural search. We already found the similar rule for WikiText-103 dataset (not shown), although the magic number (120 for Baby10M) seems different depending on datasets. If this rule holds for general datasets, when you search for the best model for a new dataset, probably you can start finding the magic number of the rule for that dataset first. Once the prospected total number of heads can be estimated first, then, you can save your exploration cost quite a lot. Although "120-rule" by itself cannot select a unique best model, having another rule as well like "about 700 hidden-dim needed to represent a token in FFNs" could uniquely determine. Typically, language models have not only attention structures but also feed forward networks. The balance between the (input) dimensions of attention structures and FFNs may be the key for the best performance like two wheels.

Limitations

Generalizability is unclear. So far, other than Baby10M, Baby100M and CIFAR10, we have observed the significant benefits of the FORGETTER training algorithms only for WikiText-103 and WikiText-2 as training datasets (not shown). It is highly important to examine how general the benefit is by trying different (possibly large) training datasets.

Only GPT Tokenizer was used. We are not sure if there is a better one. Although we almost did not explore alternative tokenizers, we hope that the comparison results, such as normal versus FORGETTER models, are general to some extent. Also, it is not clear if an existing tokenizer like the GPT 2 Tokenizer is biologically plausible. Maybe tokenizer should also be learnt from BabyLM with limited vocabrary. We need further study.

We entirely used 512 as a input token length throughout the paper. Although this length is shorter than that of GPT2 (=1024), we observed that the effect on the performance is mild, compared with shorter input lengths such as 256 or 128. However, scalability to long text should be checked with large GPU resources.

Transfer learning (instruction learning) is nowadays important for LLMs. Then the effect of the FORGETTER, that was used for pretraining, on fine-tuning is interesting. Our evaluation was by next token prediction throughout this paper. Therefore it is interesting how the FORGETTER learning beyond the normal overfitting limits can affect the following instructive learning. Relatedly, distillation is nowadays important for small language models. The combination of FORGETTER with distillation is interesting but to be done.

Although we repeated training beyond the normal overfitting limit as a curriculum learning, we just repeated the same type of learning homogenously. It is possible a model is good at some topic but not in another. By sampling training data from the topics the model is not good at, you could accelerate the training (Müller et al., 2025).

Ethics Statement

This work complies with the ACL Ethics Policy.

Acknowledgements

KM is partially supported by JSPS KAKENHI Grant Number JP25K15283.

References

Rowel Atienza. 2020. *Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more*. Packt.

Francois Chollet. 2021. *Deep Learning with Python 2nd Edition*. Manning.

Martin Ford. 2018. *Architects of Intelligence: The truth about AI from the people building it*. Packt Publishing.

Google DeepMind Gemma Team. 2024. Gemma 2: Improving open language models at a practical size.

Aurlien Geron. 2022. *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. The MIT Press.

M Konishi, KM Igarashi, and K Miura. 2023. Biologically plausible local synaptic learning rules robustly implement deep supervised learning. *Front Neurosci.*, 17:1160899.

TP Lillicrap, A Santoro, L Marris, CJ Akerman, and G Hinton. 2020. Backpropagation and the brain. *Nat Rev Neurosci.*, 21(6):335–346.

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. 2025. Small language models: Survey, measurements, and insights.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540.

Reuven Müller, Ying Xie, Linh Le, and Shaoen Wu. 2025. Dynamic knowledge elicitation: Leveraging student feedback for improved language model distillation. *Proceedings of International Joint Conference on Neural Networks 2025 (IJCNN2025)*.

H Norimoto, K Makino, M Gao, Y Shikano, K Okamoto, T Ishikawa, T Sasaki, H Hioki, S Fujisawa, and Y Ikegaya. 2018. Hippocampal ripples down-regulate synapses. *Science*, 359(6383):1524–1527.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Sebastian Raschka. 2024. *Build a Large Language Model (From Scratch)*. Manning.

Denis Rothman. 2024. *Transformers for Natural Language Processing and Computer Vision - Third Edition: Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3*. Packt.

Fares Sayah. 2022. Cifar-10 images classification using cnns (88%).

Leslie N. Smith. 2017. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472.

Evin Tunador. mingemma (github).

Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. 2022. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O'Reilly.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. Galore: memory-efficient llm training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Large Language Models and Children Have Different Learning Trajectories in Determiner Acquisition

Olivia La Fiandra*
University of Maryland, College Park
olivia.lafiandra@gmail.com

Nathalie Fernandez Echeverri*
University of California, Berkeley
nfe@berkeley.edu

Patrick Shafto
Rutgers University, Newark
patrick.shafto@gmail.com

Naomi H. Feldman
University of Maryland, College Park
nhf@umd.edu

Abstract

Large language models are often compared to human learners based on the amount of training data required or the end state capabilities of a learner, yet less attention has been given to differences in their language learning *process*. This study uses determiner acquisition as a case study to characterize how LLMs and children differ in their learning processes. By analyzing annotated speech samples from specified age ranges of four children and intermediate training checkpoints of the Pythia-70m language model, we trace the learners' learning paths of definite and indefinite determiner use. Our results reveal a divergence: the children first produce the indefinite determiner, while the model first produces the definite determiner. This difference reflects underlying differences in the learning goals and mechanisms of models and children. Framing language learning as movement over distributions of linguistic features makes the learning process visible and offers an alternative approach for comparing humans and language models.

1 Introduction

Researchers have often looked to human language learning to quantify progress in language modeling. However, most of the existing evidence for these comparisons comes from two sources: sample efficiency and end-state benchmarks. Sample efficiency measures the linguistic input required to learn language. Large language models (LLMs) are considerably less sample efficient than human language learners. While a child of age 12 will hear less than 100 million tokens in their language environment, language models are trained on data containing billions to trillions of tokens (Warstadt et al., 2023). To compare how language models' learning compares to that of human learners, researchers often rely on benchmarks that characterize the end state of the model. For example, LLMs now achieve high accuracy on evaluating grammatical well-formedness (Papadimitriou et al., 2022), yet they still show weakness in handling certain syntactic dependencies compared to humans (Marvin and Linzen, 2018). Additionally, some models fail to generalize grammatical knowledge to novel contexts that require knowledge of structural relationships (e.g. the relationship between the subject and object of a verb) (Wilson et al., 2023). Although these approaches clearly demonstrate that models are different from humans, they provide little insight into what is different in the *learning process*. Examining the learning process itself could reveal possible disparities between models and humans such as whether linguistic knowledge is acquired under different initial conditions, at different speeds, in different orders, or with varying consistency. How best to quantify these differences remains an open question.

In this paper, we take a new approach to characterizing the differences in learning between LLMs and humans, by looking at the learning trajectory for the acquisition of determiners. Specifically, we examine the definite article *the* and the indefinite article *a* that occur before a noun to specify its referent.

While prior work on language models has focused on sample efficiency and end-state benchmarks, comparing model behavior to child language acquisition provides a complementary perspective on models' determiner acquisition. Some elicitation studies on children's determiner acquisition suggest that children overuse *the* in indefinite contexts (Wexler, 2011; Maratsos, 1976). However, other developmental research shows that children acquire singular definite determiners like *the* rapidly and in adult-like ways in naturalistic production from as early as 1.5–2 years of age (Ying et al., 2024). These findings suggest that children's determiner acquisition is guided by both linguistic input and emerging pragmatic competence, high-

*These authors contributed equally to this work.

Proceedings of the First BabyLM Workshop, pages 100–108
November 8, 2025

lighting the importance of examining learning trajectories rather than just end-state performance. Including child data allows us to situate model learning in relation to human acquisition and provides a benchmark for evaluating not only what models learn, but also how learning unfolds over time.

In our data, we find that children first produce the indefinite article, whereas the model we test first produces the definite article. This difference is not only about which forms are produced, but also about the order and pattern of acquisition over time. By analyzing these trajectories, we can see that models and children may prioritize different aspects of language and acquire determiners in different sequences. We argue that this divergence reflects fundamental differences in how LLMs and children approach language learning, offering insight into the mechanisms underlying the language learning process.

2 Methods

To investigate the way language models build their linguistic knowledge, we use determiner acquisition as a case study. We define determiner use as a multinomial distribution where for each determiner phrase produced, one of three events can occur: a definite article like *the* is used, an indefinite article like *a* or *an* is used, or the required determiner is omitted.

We annotated samples of both children's and a model's determiner use, detailed in Sections 2.1 and 2.2, to trace each learner's learning trajectory. We outline the annotation processes for the child and model data in Section 2.3. We define a learning trajectory as the distributional shifts in determiner usage over time. For example, a learner might initially omit all determiners and then progress to a roughly equal distribution of definite and indefinite determiners. The learning trajectory captures this shift from the initial to the final distribution by tracing distributions of determiner use throughout the acquisition process.

To visualize a learning trajectory, we plot points representing the learner's determiner use on a simplex which maps three points on a 2D triangular plane. Each trajectory begins with the learner's initial distribution of determiner use and progresses toward a defined target distribution, specified separately for the child and model data in Sections 2.1 and 2.2, respectively. Figure 1 illustrates the trajectory of a learner who starts with an equal use of the definite, indefinite, and omitted determiners and gradually shifts toward a balanced distribution of definite and indefinite determiners.

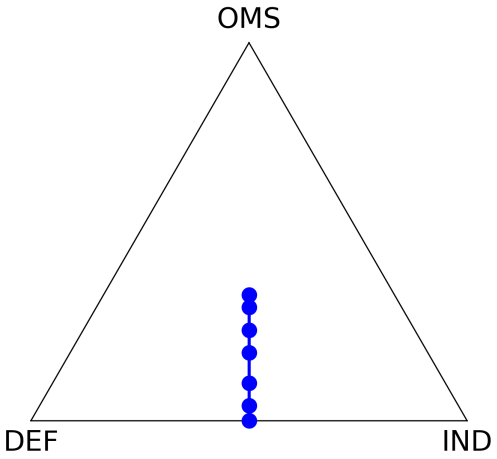


Figure 1: Learning trajectory moving from one-third event distribution to equal distribution of definite and indefinite determiners.

2.1 Child Data

We sampled the child data from the Braunwald and Providence Eng-NA CHILDES corpora (Braunwald, 1997; Demuth et al., 2006; Fernandez et al., 2024). These samples were taken from speech between children and adults, so the children's input is adult speech. We annotated cases where a required determiner was omitted in samples of four children's speech throughout early childhood. For Child 1, we annotated a sample of 4443 lines that spanned across the child's early childhood from 18 months-old to 40 months-old. For the other three children, we annotated 6 samples of their speech, one sample for each age range. For these three children's samples, we aimed to annotate 100 determiner uses per sample, although, in some samples, fewer than 100 determiner uses occurred. We also annotated samples of each child's parent's speech. We sampled and annotated the parent's speech in the same way as we did the children's speech. To check the reliability of the annotations, two annotators independently annotated a sample of the data, yielding a Kappa score of 0.99. One of the two annotators, whose reliability we measured, annotated the child data.

After we annotated the data, we used the determiner counts to plot the children's learning trajectories. We used the children's determiner distri-

butions at the 18.0–22.0 age range as their initial determiner distributions. We defined the children's target distributions based on the distributions calculated from their parents' determiner use.

2.2 Model Data

The model used throughout this study is the Pythia-70m model from the Pythia Suite (Biderman et al., 2023). We chose to use a Pythia model because it contains 154 intermediate training checkpoints. This allows us to probe the model's language use throughout training. Pythia-70m is an autoregressive causal model trained to predict the next token given all previous tokens in the context. Pythia-70m was trained on the Pile dataset. The model saw approximately 300 billion tokens in total, and each intermediate checkpoint of the model processed a batch of approximately 2 billion tokens. These checkpoints include a checkpoint at every 1000 training steps from 0 to 143000. Additionally, the checkpoints include 10 log-spaced checkpoints ranging from 1 to 512. We sampled linguistic output from the following checkpoints: 128, 256, 512, 1000, 2000, 3000, 4000, 5000, 10000, 17000, 35000, 53000, 71000, 107000, and 143000.

To parallel the input children received, we prompted the model with adult speech. Just as children hear and respond to adult speech, the model's input consisted of adult utterances. For every checkpoint we tested, the model received the same 100 lines sampled from a parent's speech in the Braunwald corpus. These lines were randomly selected and contained more than three words. After each prompt, the model generated a response of up to 20 tokens using a greedy-decoding strategy, selecting the token with the highest probability at each step. Each response included a repetition of the prompt followed by the newly generated tokens. This design ensures that the differences we observe between child and model trajectories reflect the learners' processes rather than characteristics of the input.

After counting the uses of determiners at each checkpoint, we calculated the learning trajectory. We used the determiner distribution at checkpoint 128 as the initial determiner distribution. We chose this as the initial distribution because checkpoint 128 was the earliest checkpoint to consistently produce language. The target distribution used to determine the learning trajectory of the model was the model's distribution of determiner use after training. This is the distribution found at the final checkpoint, checkpoint 143000.

2.3 Annotations

To find the learning trajectories of the model and children, we annotated the samples to find the learners' productions of definite, indefinite, and omitted determiners. This study tracked the definite determiner *the* and the indefinite determiners *a* and *an*. We did not track other determiners like *these* and *which*.

We annotated the children's determiners in the stem and gloss fields included in the CHAT format of the corpora. The stem field corresponds to the base form of the utterances, while the gloss field corresponds to the utterances' intended meanings. We annotated determiner omissions by marking the omission with a *0* preceding the omitted determiner in the stem field and the determiner occurring in the gloss field. We annotated appropriate determiner uses with the determiner occurring in both the stem and gloss fields. Table 1 illustrates examples of these annotations.

Next, we annotated the model's output for determiner use, tracking definite, indefinite, and omitted determiners across its linguistic output. Because the model's production was restricted to 20 tokens, it occasionally cut off its response on a determiner. This type of determiner use was annotated as an End of Response Use. Additionally, the model occasionally cut itself off and started a new line of its response. In the cases where the model cut itself off on a determiner and started a new line of its response, we annotated these determiners as Cut Off Uses. We counted End of Response Uses and Cut Off Uses toward the total number of determiners produced by the model in order to accurately calculate the distributions of determiner uses. This is because these incomplete productions still indicate the model's choice to use a determiner in that context, and excluding them would underestimate the determiner frequency. Because the model first repeated the prompt and then generated new tokens in its response, the model may repeat determiners from the prompt in its response. We did not count these determiner occurrences as determiner productions by the model. Table 2 contains examples of these annotations.

For both the model and the children, each consecutive use of the same determiner counted toward the total definite and indefinite uses. For example,

Type	Use	Stem	Gloss
Definite	I saw the dog	I saw the dog	I saw the dog
Indefinite	She is not a toy	She is not a toy	She is not a toy
Omission	It has book inside it	It has 0a book inside it	It has a book inside it

Table 1: Examples of annotations for the child data.

Prompt	Response	Definite	Indefinite	Omission	End of Response	Cut Off
It's a dangerous toy if you can't abide by the rules	It's a dangerous toy if you can't abide by the rules. "I'm not going to be a toy that I'm not going to be a"	0	2	0	1	0
Okay here you go	Okay here you go to the "I'm not sure what you're going to do," he said. "	1	0	0	0	1

Table 2: Examples of annotations for the model data.

the the ball counts as two definite determiner uses.

3 Learning Trajectories

Figure 2 shows the learning trajectory of the model and the average learning trajectory of the children, accompanied by the children's individual trajectories for reference. We assume that before any learning occurs, children would produce a 100% omission distribution, because without having learned determiners, they would never use one in a required context. This represents an unobserved portion of the trajectory, with a hypothetical point at 100% omissions preceding our first measurement. We do not assume the same hypothetical point for the model, and instead expand on the model's pre-learning distribution in Section 4.

At the first measured point, the model exhibits a 100% definite, 0% indefinite, 0% omission distribution. Over the course of training, the model shifts toward a more balanced distribution of the definite and indefinite determiners while never producing omissions. The model's trajectory indicates that the model first produces the definite determiner before the indefinite determiner. In contrast, the children's initial distributions favor the indefinite determiner over the definite, showing that they produce the indefinite determiner first. These differences in initial determiner production reflect that the model and children acquire determiners in different sequences, and this difference in order of acquisition indicates that their learning processes operate differently.

Despite these different sequences of acquisition, both the model and children converge near approximately equal distributions of definite and indefinite determiners by the end of their trajectories. This convergence suggests that despite differing orders of acquisition, both the children and the model ultimately reach a similar endpoint in determiner use. This highlights that different learning processes can produce comparable outcomes, and that the learning trajectory may be an informative measure of language learning.

One way to interpret the comparison is to focus on the portion of the trajectory after children start leaning toward determiner production instead of omission. From this perspective, the model and children behave similarly in how they approach the target distributions, and this can be quantified by measuring their *distance* from the target at each point in time. This measure captures how far a learner's determiner use is from the expected distribution and allows us to examine how quickly that distance decreases relative to the amount of observed data. Figure 3 plots the children's distances and the average distance to the target at each age range and shows the model's distance to the target at each checkpoint on a log scale. The distances were calculated using KL divergence. These figures illustrate how the gap between learner output and the target narrows over time in both the model and children, while also showing that the model's trajectory accelerates more quickly over the course of learning than the children's. Examining whether the trajectory accelerates or slows at different points provides a way to compare learning patterns.

The behavior of the model first producing the definite determiner and then the indefinite determiner can potentially be explained based on the model's language objectives. The objective of

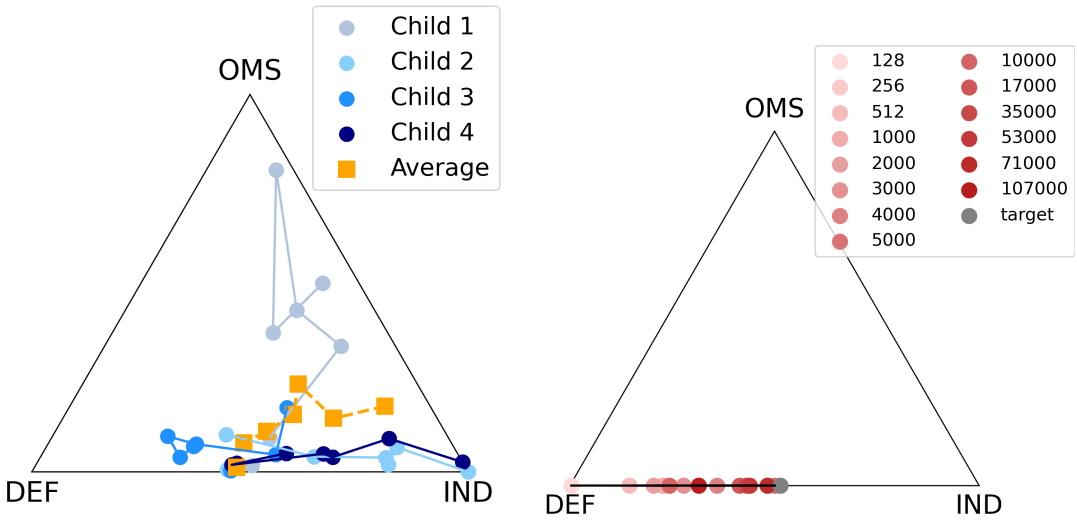


Figure 2: Children's individual and average trajectories (left) and model's learning trajectory (right).

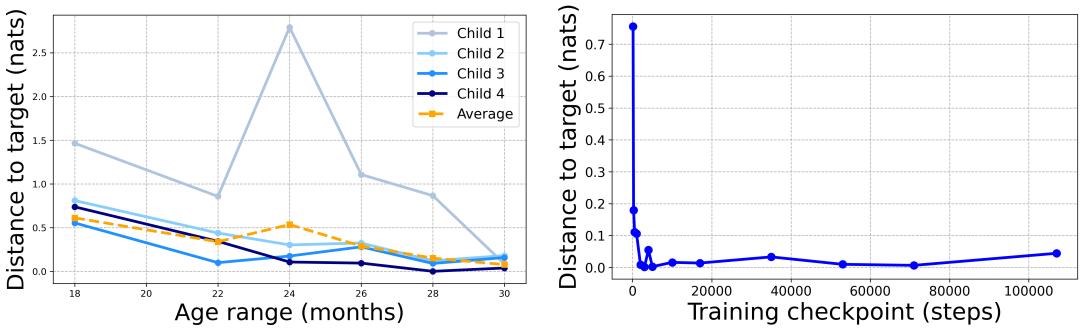


Figure 3: Children's individual and average distances from target at each age range (left) and model's distance from target at each training checkpoint.

Pythia-70m is to generate language by predicting the next word based on the linguistic context. To do this, the model generates the word with a high probability of occurring next based on the prior linguistic context. In early stages of training, the model may generate the definite determiner more often than the indefinite determiner due to the high frequency of the definite determiner occurring in language. For example, data from the Corpus of Contemporary American English (COCA) show that *the* occurs roughly twice as often as *a* or *an* (Davies, 2008). It may be the case that, when learning from the frequencies in its input, the model may initially gravitate toward the more frequent option before it has enough exposure to produce the indefinite determiner.

What this does not explain is why the model does not probability match by the end of training. We should not expect to see the model reach an equal distribution of definite and indefinite determiners at the end of training if the definite determiner is more likely to occur than the indefinite determiner in its training data. Though this is puzzling, it is not inconsistent with the children's target distributions which also approximate equal distributions of the definite and indefinite determiners. This pattern suggests that factors other than data frequencies may contribute to shaping the model's final determiner distribution.

While the model's initial behavior aligns with the fact that the frequency of the definite determiner is higher than the frequency of the indefinite determiner in language input, the children's behavior does not. Despite the definite determiner occurring twice as often as the indefinite determiner in language input, children appear to first produce the indefinite determiner before they produce the definite determiner. One possible reason for this could be that it is easier for children to grasp the meaning of the indefinite determiner than the definite determiner. The meaning of the definite determiner is connected to concepts of uniqueness and

specificity. There is a difference between saying *I have the ball* and *I have a ball*. *The ball* refers to a specific object that is salient to the speaker. It may be the case that children do not yet understand these concepts. Supporting this, a study on infants' perspective-taking in language comprehension found that 14-month-olds do not demonstrate an understanding of specificity, while 19-month-olds do (Choi et al., 2018). In this experiment, two agents and a participant interacted with an apparatus containing two identical balls. The participant and Agent 1 could see both of the balls while Agent 2 could only see one of the balls. When Agent 2 requested *the ball* from Agent 1, 19-month-olds expected Agent 1 to hand over the specific ball visible to Agent 2, showing sensitivity to the uniqueness and saliency of the ball. In contrast, 14-month-olds accepted either ball as a valid referent. This developmental gap between 14-month-olds and 19-month-olds suggests that younger children have not yet grasped concepts of uniqueness and specificity that are required to appropriately use the definite determiner. A lack of understanding of uniqueness, despite a high frequency of the definite determiner in language input, offers a possible explanation for why children's production patterns diverge from frequency-based expectations.

4 Qualitative Analysis

To better understand the model's early behavior, we conducted a qualitative analysis of its determiner use at early training checkpoints. This approach allows us to examine unexpected patterns that quantitative measures might not capture. One such pattern is the model's 100% definite distribution at the first checkpoint we measured.

To investigate the model's frequent production of the definite determiner in early stages of training, we turned to examine the content of the model's responses at checkpoint 128. We found that the model never appropriately uses either determiner despite frequently producing the definite determiner. This is because after repeating the prompt, the model looped its responses ending with determiners. Table 3 demonstrates some of these responses.

The model's behavior at checkpoint 128 leads us to ask why the model repeats the determiner and loops phrases like *and the* that end in a determiner. One possible explanation for this behavior is that the words that the model loops are the only words the model has learned at that point. This would explain why the model only generates the same words—specifically the words *and* and *the*. This explanation also suggests that these are the first words the model learns throughout training. Under this explanation, the model's distribution of determiner use prior to our observing its behavior may not exist. If the first words the model produces are *the* and *and*, then there is never a time when the model omits a required determiner before determiner-learning begins. This is because language learning for the model begins with learning the definite determiner. While the model behaves this way early on in training, we do not see the same behavior from the children. This qualitative difference again suggests that the model and children learn determiners in different ways.

5 General Discussion

This study provides a comparison of determiner acquisition between children and a large language model, highlighting how learning trajectories reveal differences in the sequencing of language learning. Our results show that the model and children differ primarily in where each begins its trajectory: the model initially exhibits a strong preference for the definite determiner, whereas children start with a higher proportion of indefinite determiner use. Over time, both learners converge toward a roughly balanced distribution of definite and indefinite determiners. These findings go beyond prior work by examining not just the end state of language learning or the amount of linguistic input needed to learn, but also the learning process itself, illustrating how sequences of acquisition can differ across humans and models.

While we assume that children begin with a relatively high level of omissions of determiners and learn toward the indefinite determiner once determiner-learning begins, we find that the model begins with 100% use of the definite determiners. These differing starting points raise questions about the underlying mechanisms driving early behavior. Two possible ideas explored in Sections 5.1 and 5.2 could account for this divergence.

In addition to these differences in starting points, a factor shaping the model's learning trajectory is the decoding strategy used. We used greedy decoding, the model's default decoding strategy, which deterministically selects the most probable token at each step. This approach may exaggerate early pref-

Prompt	Response
okay here you go	okay here you go, and the, and the the the the the the the the the the the the the the the the
we're recording our voices	we're recording our voices and the the the the the the the the the the the the the the the the the the the the the
I did wanna hear why don't you sing it together	I did wanna hear why don't you sing it together, and the, and the, and the, and the, and the, and the, and

Table 3: Examples of responses at checkpoint 128.

erences, such as the model's strong initial bias toward the definite determiner. Alternative decoding strategies could have produced more variable determiner distributions, potentially smoothing or delaying the observed trajectory. While our findings show how determiner use unfolds under greedy decoding, future work should explore whether other decoding strategies alter the model's trajectory or reveal additional stages of learning.

Another observation from our findings is the disconnect between the child data and prior developmental work noted in Section 1. While our results show that children's productions include relatively frequent use of the indefinite determiner early on, prior developmental research has demonstrated children's tendency to overuse definite determiners in experimental contexts. One way to reconcile this discrepancy is to consider differences in experimental and naturalistic contexts. Experimental tasks may introduce pressures that favor definite determiners, whereas naturalistic production data, like the data analyzed here, capture children's baseline use more directly. This comparison emphasizes the importance of context in shaping conclusions about early determiner use and highlights the value of including child data in our comparison with the model. The child data not only illustrates how the model diverges from human learners, but also reveals how different methodological perspectives can yield distinct views of children's developmental trajectory.

5.1 Children and Models Are Different Types of Learners

One possible explanation for the difference between the model and children's learning processes is that children and models are different types of language learners. The model used in this study generates language by predicting the next word based on the probability of the word occurring in a specific context. The model forms these probabilities based on the frequencies of words in its training data, per its training objective. Because the definite determiner occurs frequently in English, the model may have a preference for learning it before learning the indefinite determiner. This would explain why the model starts its learning trajectory with a higher probability of using the definite determiner than using the indefinite determiner. In contrast, these same frequency distributions in children's linguistic environments may not be sufficient on their own to make it easier for children to produce the definite determiner, given underlying conceptual challenges involved in its appropriate use (Arunachalam and Waxman, 2010; Booth et al., 2005). One such conceptual challenge may be understanding the concepts of uniqueness and specificity, discussed in Section 3. If it is the case that language learning is impacted by conceptual challenges, then children may not be able to overcome the challenge of appropriately using the definite determiner over the indefinite determiner using frequencies alone.

5.2 Children and Models Use Language Differently

Another explanation for the difference between learning trajectories of the model and children is that language models and children use language differently. The objective of our language model is to generate language by predicting the next word. In order to complete this goal, the model chooses the word with a high probability of occurring next in the sentence. The goal of a child is different from the goal of a language model. A child's goal is not to predict the next word. Rather, their goal is to communicate (Tomasello, 2003). One possible explanation which would require further research

is that children may tolerate certain errors, including omissions of determiners, in order to efficiently communicate. For example, research has found dissociation between production and comprehension in language use, suggesting that speakers sometimes produce ungrammatical utterances because they cannot retrieve the appropriate form in the moment (Harmon and Kapatsinski, 2017). This could explain why the children in our study omit determiners. The meaning of a child's message is likely not affected by the omission of a determiner, so children may tolerate those mistakes depending on other production challenges they face. For example, a child asking for their bottle may correctly say *I want the bottle* or incorrectly say *I want bottle*. In this case, though omitting a determiner is ungrammatical, the omission does not impact the meaning of the message the child is attempting to communicate.

6 Conclusion

This study examined differences in language learning between models and children by exploring their learning trajectories. We found that models and children behave differently throughout determiner acquisition in regards to their learning processes. The model and the children appear to produce the definite and indefinite determiners in opposite order: the model beginning with the definite determiner and the children beginning with the indefinite determiner. These learning differences reflect disparity in the goals and learning processes that shape language models and humans. It appears that next-word prediction objectives and probabilistic optimization drive determiner use for the model, while communicative needs and learning milestones drive determiner use for children.

The approach of framing acquisition as movement over distributions makes the underlying process of acquisition visible and provides a nuanced basis for comparing language models to human learners. Learning trajectories provide a way of measuring *how* the learning process unfolds for different learners. This study shows a difference in the sequencing of determiner acquisition in models and children, and additional data could reveal further differences such as differences in speed or consistency. Future work could also extend this approach to other aspects of grammar and work to build a broader map of learning differences between models and humans.

By comparing models and humans through their learning trajectories rather than end state learning or the quantity of training data, this approach uncovers fundamental differences in how LLMs and humans learn language over time. Comparing learning processes makes visible the paths and intermediate steps of language learning which offers insights into the mechanisms driving the learning process. This perspective emphasizes the importance of studying *how* learning unfolds, not just *what* is learned or *how much data* is needed. Ultimately, this process-oriented approach provides an alternative way to evaluate and improve LLMs.

Limitations

While this study provides insights into the differences between LLMs and children's determiner acquisition processes, several limitations should be noted. First, the analysis is based on a single large language model and a relatively small set of child learners. The findings may not generalize across other models with different architectures or optimization objectives, nor across child learners with varying linguistic backgrounds. Second, the study focuses exclusively on the articles *the*, *a*, and *an* which represent only a subset of English determiners. Examining a broader range of determiners could reveal different learning trajectories. Third, we measured the model's behavior through its productions, and did not probe its logits or embeddings. This limits our ability to interpret aspects of the model's behavior, such as repeatedly producing the definite determiner. While inspecting the model's generation function and examining the logits for all vocabulary items could clarify this behavior, such an analysis was not completed in this study. Therefore, the early dominance of *the* should be interpreted cautiously. Finally, greedy decoding may amplify the model's early preference for the definite determiner, and future work should examine whether alternative decoding strategies yield different developmental patterns in the model.

Acknowledgments

This research was supported by a University of Maryland Baggett Fellowship and by the University of Maryland Strategic Partnership: MPowering the State, a formal collaboration between the University of Maryland College Park and the University of Maryland Baltimore. We thank the computational cognitive science group for helpful comments and

discussion.

References

Sudha Arunachalam and Sandra R. Waxman. 2010. Language and conceptual development. *WIREs Cognitive Science*, 1(4):548–558.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Amy E. Booth, Sandra R. Waxman, and Yi Ting huange. 2005. Conceptual information permeates word learning in infancy. *Developmental Psychology*.

Susan R. Braunwald. 1997. The development of because and so: Connecting language, thought and social understanding. In *Processing Interclausal Relationships in the Production and Comprehension of Text*. Lawrence Erlbaum Associate, Hillsdale, New Jersey.

Youjung Choi, Hyun joo Song, and Yuyan Luo. 2018. Infants' understanding of the definite/indefinite article in a third-party communicative situation. *Cognition*, 175:69–76.

Mark Davies. 2008. The corpus of contemporary american english.

Katherine Demuth, Jennifer Culbertson, and Jennifer Alter. 2006. Word-minimality, epenthesis and coda licensing in the early acquisition of english. *Language and Speech*.

Nathalie Fernandez, Rose Griffin, Patrick Shafto, and Naomi Feldman. 2024. Characterizing language learning trajectories with optimal transport. In *Paper presented at the Boston University Conference on Language Development*.

Zara Harmon and Vsevolod Kapatsinski. 2017. Putting old tools to novel uses: The role of form accessibility in semantic extension. *Cognitive Psychology*, 98:22–44.

MP Maratsos. 1976. *The use of definite and indefinite reference in young children*. Cambridge University Press.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Isabel Papadimitriou, Richard Futrell, and Kyle Mahowald. 2022. When classifying grammatical role, bert doesn't care about word order... except when it matters.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Ken Wexler. 2011. Cues don't explain learning: Maximal trouble in the determiner system. *The processing and acquisition of reference*, 15.

Michael Wilson, Jackson Petty, and Robert Frank. 2023. How abstract is linguistic generalization in large language models? experiments with argument structure. *Transactions of the Association for Computational Linguistics*, 11:1377–1395.

Yuanfan Ying, Valentine Hacquard, Alexander Williams, and Jeffrey Lidz. 2024. Children do not overuse "the" in natural production. *Proceedings of the 48th annual Boston University Conference on Language Development*.

Design and Analysis of few Million Parameter Transformer-based Language Models trained over a few Million Tokens Dataset

Yen-Che Hsiao[1], Abhishek Dutta[1]

[1]Department of Electrical and Computer Engineering
University of Connecticut
Storrs, 06269, CT, USA

Correspondence: yen-che.hsiao@uconn.edu

Abstract

In this work, we systematically explore training methods and perform hyperparameter tuning to identify key language model parameters upper bounded by 28 million. These models are designed to generate a broad spectrum of basic general knowledge in simple and coherent English with limited generalization ability. We use the Simple English Wikipedia as the training dataset, selecting samples between 64 and 512 words, which provides a high-quality, compressed representation of general knowledge in basic English. Through hyperparameter tuning, we identify the best-performing architecture, yielding the lowest training loss, as a decoder-only Transformer with rotary positional encoding, multi-head attention, root-mean-square normalization, Gaussian error linear unit activation, post-normalization, no interleaved group query attention, an embedding dimension of 512, 8 layers, 8 attention heads, a feedforward dimension of 2048, and zero dropout. Models trained with a learning rate decaying linearly from 10^{-4} to 10^{-5} over 64 epochs achieve a training loss of 0.1, which appears sufficient for reproducing text more effectively than models trained to losses of 0.2 or 0.5. Fine-tuning on rephrased text further demonstrates that the model retains its ability to produce simple and coherent English covering broad basic knowledge, while exhibiting limited generalization capability.

1 Introduction

Several works have developed language models capable of performing a wide range of tasks (Sindhu et al., 2024), including but not limited to code completion (Husein et al., 2025), question answering (Nassiri and Akhloufi, 2023), and text summarization (Zhang et al., 2025). Large language models often contain more than a billion parameters and are typically trained on more than 100 billion tokens (Raiaan et al., 2024). However, such large-scale models present challenges in terms of deployment on local devices, and their substantial carbon footprint underscores the need to reduce computational requirements while maintaining comparable or acceptable performance.

Practitioners have also released language models with less than one billion parameters that are capable of generating grammatically correct and informative text, such as the 0.6B models from Qwen3 (Yang et al., 2025) and the 135M and 360M models from SmolLM (Allal et al., 2025). While some information regarding the model architecture, training hyper-parameters, and training data is often provided, details on how the architecture, hyper-parameters, model checkpoints, or datasets are selected are rarely, if ever, discussed.

In this work, we aim to identify the transformer-based language model with the minimum number of parameters less than 28 million (M) capable of generating broad spectrum of basic general knowledge in simple and coherent English and has limited generalization ability. The Simple English Wikipedia dataset is used for training, as we consider it a high-quality and compact dataset that covers a broad range of general knowledge in basic English, compared to other datasets such as WikiText-2. As a starting point, we set the target of learning a dataset containing 20 M tokens within 10 epochs. Following the result reported in Table 3 of (Hoffmann et al., 2022), we use the empirical observation that the optimal number of model parameters is approximately equal to the total number of training tokens divided by 20. This yields a target of roughly 10 M parameters for our setting (20 M tokens $\times$ 10 epochs $\div$ 20). Our study therefore focuses primarily on models with parameter counts exceeding 10M, up to 28M parameters, for over-parameterization.

We investigate models with varying numbers of decoder blocks, trained under different layers, batch sizes, learning rates, and training strategies. For dataset selection, we compare the Sim-

Proceedings of the First BabyLM Workshop, pages 109–129

ple English Wikipedia dataset with the WikiText-2 dataset. We discard WikiText-2, since it contains non-English characters as some of the most frequent tokens, suggesting that the dataset may not consist of coherent and simple English compared to Simple English Wikipedia. The Simple English Wikipedia dataset is further processed by removing entries with fewer than 64 words and more than 512 words, as shorter entries often consist of incomplete sentences and longer entries exceed the sequence length of the model. The resulting dataset is referred to as the long-context subset. Analysis of the word frequency distribution in Simple English Wikipedia shows that 2,206 words appear more than 500 times, motivating the choice of a vocabulary with 2,048 tokens.

We evaluate models trained on the Simple English Wikipedia dataset under different batch sizes, learning rates, and training strategies. The first strategy, referred to as the 2-stage training method, involves initially training on short-context data with a small maximum sequence length for several epochs, followed by training on long-context data with a larger maximum sequence length. The second strategy, referred to as the interleaved training method, alternates between short- and long-context datasets each epoch, with corresponding adjustments to the maximum sequence length, to mitigate catastrophic forgetting. Our results show that the best-performing model, achieving the lowest training loss within 10 epochs, is a decoder-only Transformer with rotary positional encoding, multi-head attention, root-mean-square normalization, Gaussian error linear unit (GELU) activation, post-normalization, no interleaved group query attention, an embedding dimension of 512, 8 layers, 8 heads, a feedforward dimension of 2048, zero dropout, and a learning rate of 10^{-4} trained with standard mini-batch gradient descent. Analysis of generated text indicates that a training loss of 0.1 is sufficient for the model to reproduce the target text more effectively than models trained to losses of 0.2 or 0.5.

To evaluate limited generalization ability, we fine-tune the 0.1-loss pre-trained model on rephrased text. The fine-tuning dataset is constructed from 100 selected entries of the long-context subset of Simple English Wikipedia. Each entry is split into two parts: the first as the context and the second as the target. The context is rephrased into three variants using ChatGPT-5. The rephrased contexts paired with the original target form new training and evaluation entries: the first and second rephrased contexts are used for the training set, the third rephrased context is used for the test set, and the original context is used for the validation set. After fine-tuning for 64 epochs, we select the model with the lowest validation loss and evaluate it on five entries each from the validation and test sets. The model successfully reproduces the target text for two of five entries in both the validation and test sets, succeeds only in the test set for one entry, succeeds only in the validation set for one entry, and fails on both sets for one entry. These results demonstrate that the fine-tuned model exhibits limited generalization ability, while maintaining the ability to generate simple and coherent English covering a broad spectrum of basic general knowledge.

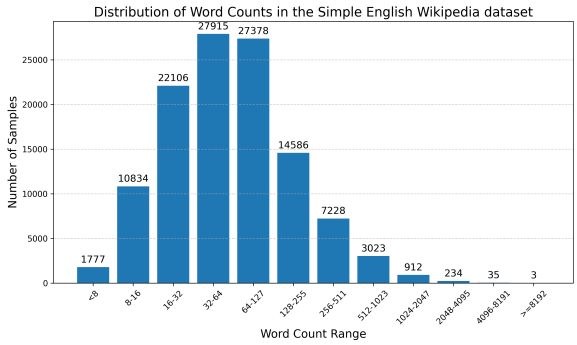
2 Data Preparation

2.1 Dataset Selection

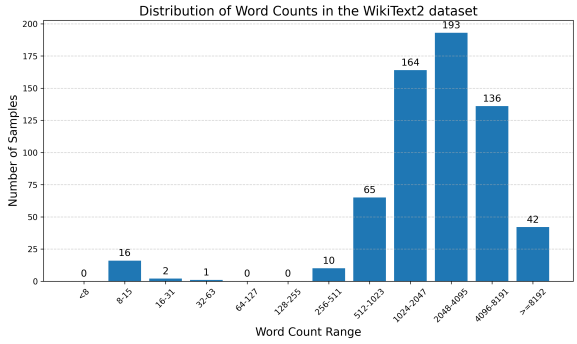
To build a model capable of generating a broad spectrum of basic general knowledge in simple and coherent English, we aim to identify a suitable text dataset for training. We considered two datasets as candidates: the WikiText-2 dataset (Merity et al., 2016) and the Simple English Wikipedia dataset from the 100M training set of the BabyLM Challenge (Charpentier et al., 2025).

To make the datasets suitable for training, we process them to ensure that each data entry corresponds to text related to a single Wikipedia topic. In the Simple English Wikipedia dataset, topics are enclosed by the "= = =" symbol, with the related context in the following lines, and each topic-context pair separated by two newline symbols. We construct the training dataset by extracting each context as a single entry, resulting in 116,031 samples containing a total of 13,707,770 words. For the WikiText-2 dataset, topics are enclosed by a single "=" symbol, while subsections and subsubsections are enclosed by "= =" and "= = =" symbols, respectively. We process the dataset by extracting the text associated with each topic enclosed by a single "=" symbol, continuing until the next topic marker. This yields 629 samples with a total of 2,051,910 words.

To compare the datasets, we first inspect histograms of sample counts versus word counts. A sample corresponds to text associated with one topic, and the word count is computed as the number of consecutive character sequences separated



(a) Simple English Wikipedia Dataset



(b) WikiText-2 Dataset

Figure 1: Histogram of the number of data entries in each word count range. Most data entries contain between 32 and 127 words in the Simple English Wikipedia dataset, while most data entries contain between 1,024 and 8,191 words in the WikiText-2 dataset.

by whitespace (spaces, tabs, or newlines). The histogram in Figure 1a shows that most Simple English Wikipedia data entries contain between 32 and 127 words, while most WikiText-2 data entries contain between 1,024 and 8,191 words, as shown in Figure 1b. We also examine the ten most frequent words in each dataset. In Simple English Wikipedia, these are: “the”, “of”, “in”, “and”, “a”, “is”, “to”, “was”, “The”, and “for”. In contrast, the top-10 words in WikiText-2 are: “the”, “,”, “.”, “of”, “<unk>”, “and”, “in”, “to”, “a”, and “=”. Four of these ten tokens are not actual words, suggesting improper processing during text extraction. Since tokens such as “<unk>” do not appear in natural English, and isolated “,” and “.” are uncommon, we conclude that WikiText-2 may degrade the ability of a language model to learn proper English. Therefore, we discard the WikiText-2 dataset.

2.2 Processing the Simple English Wikipedia Dataset

The statistics of the Simple English Wikipedia dataset from the 100M training set of the BabyLM Challenge (Charpentier et al., 2025) are presented in Table 1, where word counts are computed as the number of consecutive character sequences separated by whitespace.

To further assess dataset quality, we examine entries across different word count ranges. Examples are shown in Figure 7 to Figure 11. Entries with fewer than 8 words (Figure 7) are mostly incomplete sentences and insufficient to describe a topic. In contrast, entries with 8 words or more (Figs. 8 to 11) typically consist of complete sentences and are adequate to describe a topic. Most entries fall within 8 to 511 words, consistent with the distribution shown in Figure 1a.

2.3 64–512 Word Subset of the Simple English Wikipedia Dataset

We construct a 64–512 word subset of the Simple English Wikipedia dataset by excluding entries with fewer than 64 words. The statistics of this subset are reported in the third row of Table 1. Each sample is padded or truncated to a maximum of 1,024 tokens. Note that these statistics are computed on the truncated entries. During tokenization and detokenization, isolated symbols such as “,” and “.” are concatenated to the preceding word, resulting in slightly lower word counts than the predefined minimum.

2.4 Tokenizer

For training on the Simple English Wikipedia Dataset, we construct a tokenizer using GPT-2 style byte-pair encoding (BPE) (Radford et al., 2019) trained on the 64–512 word subset. The tokenizer includes the special tokens `[PAD]`, `[UNK]`, `[MASK]`, and `[EOS]`.

To determine the vocabulary size, we plot the frequency distribution of unique words in the 64–512 word subset (Figure 2). We observe that 2,206 unique words occur more than 500 times, motivating the choice of a 2,048-token vocabulary, which is close to this number. We also examine the effect of vocabulary size on training loss. Results show that a 4-layer model with a larger vocabulary size tends to yield higher training loss (Figure 12 in Appendix B).

3 Language Model Architecture and Hyperparameters

The architecture of the transformer-based language model (Vaswani et al., 2017), including the first

Table 1: Statistics of word counts for the Simple English Wikipedia dataset and its subsets.

Dataset	*Mean*	*Median*	*Maximum*	*Minimum*	*Total words*	*Samples*
Simple English Wikipedia	118.14	58	9,423	1	13,707,770	116,031
64–512 Word Subset	183.29	126	660	61	9,654,100	52,671

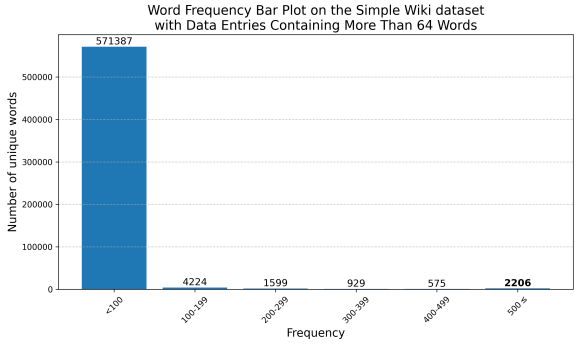


Figure 2: Bar plot of the number of unique words versus their frequency of occurrence in the 64–512 word subset of the Simple English Wikipedia dataset.

decoder block, is shown in Figure 3. The feed-forward network consists of 2,048 neurons, the attention mechanism uses 8 heads, and no dropout is applied during training. Other parameters are selected based on the experiments described in Appendix B. We found that the best-performing configuration is a decoder-only Transformer with rotary positional encoding, multi-head attention, root-mean-square normalization, GELU activation, post-normalization, no interleaved group query attention, an embedding dimension of 512, 8 layers, 8 heads, a feedforward dimension of 2,048, zero dropout, trained with a batch size of 2 and an initial learning rate of 10^{-4}.

We also observe that small models struggle to generate coherent English and reproduce factual knowledge for training samples with long contexts. To address this, we evaluated two training strategies. The first is a two-stage approach, where the model is first trained on short-context data and then on long-context data. Although this strategy produces coherent sentences, it fails to preserve factual knowledge. The second strategy interleaves short- and long-context datasets, alternating maximum sequence lengths each epoch. This mitigates catastrophic forgetting of short-context samples and produces coherent sentences, but similarly fails to preserve factual knowledge in long-context samples. The detailed results and analysis are presented in Appendix C.

4 Training

Given a set of training data $\mathcal{D} = \{d^1, d^2, \ldots, d^N\}$ with N samples, where the i-th sample $d^i = (d^i_1, d^i_2, \ldots, d^i_{k^i})$ contains k^i number of tokens, the loss for each sample is computed by the negative log-likelihood:

$$\mathcal{L}(\theta; d^i) = -\sum_{j=1}^{k^i} log(p_\theta(d^i_j | d^i_{j-1}, \ldots, d^i_1)), \quad (1)$$

where $p_\theta(d^i_j | d^i_{j-1}, \ldots, d^i_1) \in [0, 1]$ is the probability assigned by the transformer-based language model parameterized by θ to token d^i_j, given the preceding tokens $d^i_1, \ldots, d^i_{j-1}$.

The model is trained using standard mini-batch gradient descent as detailed in Algorithm 1.

Algorithm 1 Mini-Batch Gradient Descent

Input: Initial learning rate $\alpha \in \mathbb{R}$, momentum factors $\beta_1 \in \mathbb{R}$ and $\beta_2 \in \mathbb{R}$, weight decay factor $\lambda \in \mathbb{R}$, $\epsilon \in \mathbb{R}$, batch size b, maximum epochs E, dataset $\mathcal{D} = \{d^i\}_{i=1}^N$, loss function $\mathcal{L}$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$, time step $t \leftarrow 0$
Output: Optimized parameters θ
Initialize parameters θ randomly
for $ep = 1$ to E **do**
  Shuffle dataset $\mathcal{D}$
  for each mini-batch $\mathcal{B} \subset \mathcal{D}$ of size b **do**
    $t \leftarrow t + 1$
    $\eta_t \leftarrow$ SetScheduleMultiplier(t)
    Compute gradient:

$$\vec{g} \leftarrow \frac{1}{(\sum_{d^i \in \mathcal{B}} |d^i|)} \sum_{d^i \in \mathcal{B}} \nabla_\theta \mathcal{L}(\theta; d^i)$$

    Update parameters:

$$\theta \leftarrow \text{AdamW}(\vec{g}, \alpha, \beta_1, \beta_2, \epsilon, \lambda, \eta_t)$$

  end for
end for
return θ

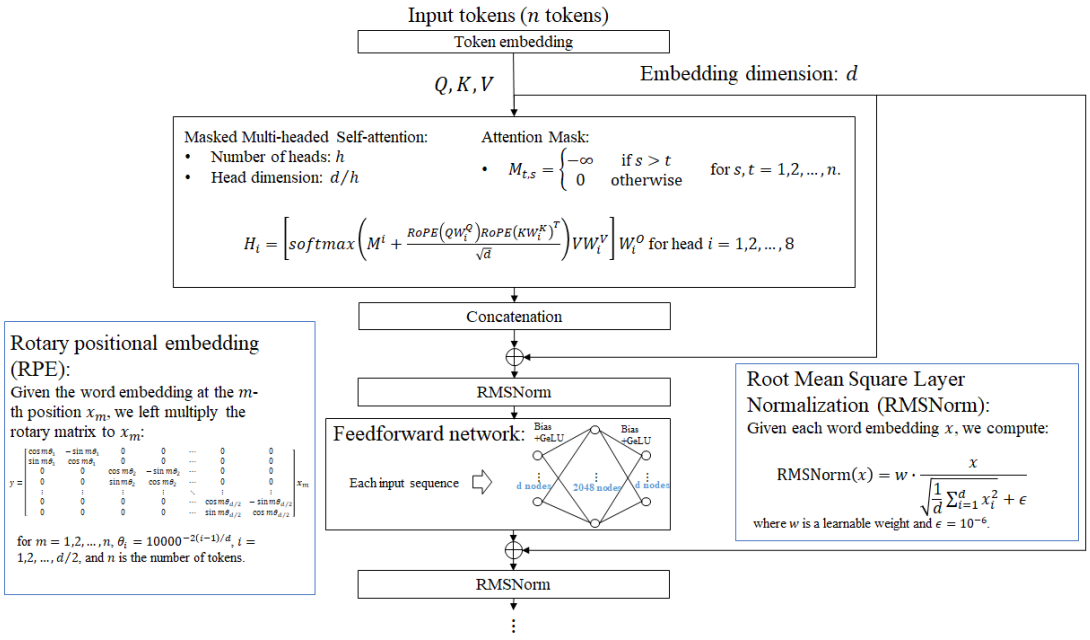


Figure 3: Illustration of the architecture of the initial part of the transformer-based language model used in this study, including the first decoder block.

The common training hyperparameters for all the experiments are $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $\lambda = 0.01$ for the AdamW optimizer (Loshchilov and Hutter, 2019) and a maximum gradient norm of 1 for gradient clipping.

4.1 Results for Training on the 64–512 Word Subset

We train the 8-layer decoder-only transformer model described in Figure 3 with an embedding dimension of 512, a vocabulary size of 2,048 tokens, and 8 attention heads, using an initial learning rate of 10^{-4} with a linear scheduler. The training loss and learning rate under two different linear scheduling strategies are shown in Figure 4a and Figure 4b, respectively. As shown in Figure 4a, the model trained with a learning rate schedule that decays 10-fold every 64 epochs achieves a training loss below 0.1 at 169 epochs, whereas the model trained with a 10-fold decay every 128 epochs reaches the same threshold at 179 epochs. The blue dotted and dash-dotted lines indicate that the model with a 64-epoch decay reaches training losses below 0.5 and 0.2 at 58 and 94 epochs, respectively.

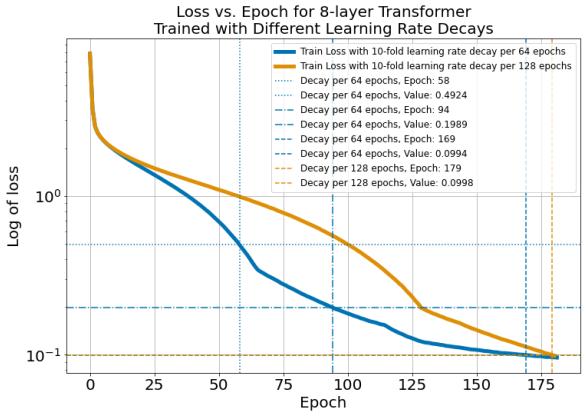
To evaluate the quality of model outputs across training stages, we focus on the model with 10-fold decay every 64 epochs, as it reaches a training loss below 0.1 faster than the model trained with a learning rate of 10-fold decay every 128 epochs. We select checkpoints corresponding to training losses of 0.5, 0.2, and 0.1, and inspect their outputs on selected training samples in Table 7 in Appendix D.

For reproducibility, Table 7 shows that the model trained to a loss of 0.1 successfully reproduces text from all three selected training entries (labeled "seen"). In contrast, the models trained to losses of 0.2 and 0.5 reproduce two of the three entries but fail on the third, suggesting that a threshold of 0.1 may be appropriate for the model to reliably reproduce training text.

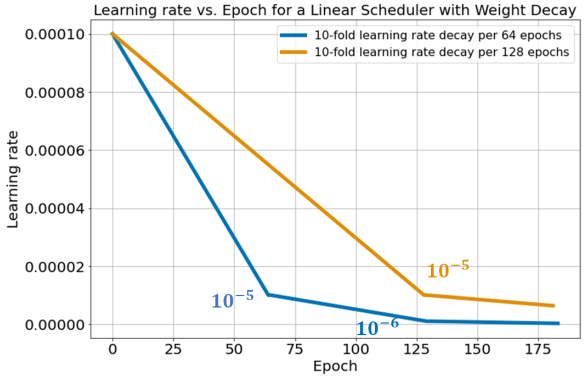
For generalization, the "rephrased" rows in Table 7 show that none of the models fully reproduce the target text when inputs are rephrased versions of the training samples. However, in the second rephrased example, the models trained to losses of 0.1 and 0.2 generate the correct date of Kahn's death, and the model trained to a loss of 0.5 even produces the correct age at death. The rephrased inputs were generated using ChatGPT-5 with the prompt: "Rephrase the following sentences with minimum changes: *input text*." These results suggest that generalization is difficult to achieve during pretraining. In the next section, we attempt fine-tuning on rephrased text to assess whether this improves generalization to unseen rephrasings.

5 Generalization

In this section, we aim to enable the model to generate coherent English for text that is rephrased from, but not present in, the training set. To build a



(a) Training loss on the 64–512 word subset



(b) Learning rate for training on the 64–512 word subset

Figure 4: Training loss and learning rate for an 8-layer transformer trained on the 64–512 word subset of the Simple English Wikipedia dataset. The blue lines correspond to a linear scheduler with an initial learning rate of 10^{-4} and a 10-fold decay every 64 epochs. The orange lines correspond to a 10-fold decay every 128 epochs. Dashed lines mark the epochs and loss values where each model reaches a training loss below 0.1. The dotted and dash-dotted blue lines mark the epochs where the model with 64-epoch decay reaches training losses below 0.5 and 0.2, respectively.

dataset for fine-tuning, we select 100 data entries with word counts between 64 and 127 words from the Simple English Wikipedia. Each entry is manually split into two parts: the first half as the context and the second half as the target. The context of each entry is then rephrased in three different ways using ChatGPT-5 with the following prompt:

> Rephrase the text in the following dictionary in 3 different ways and fill them in textR1, textR2, and textR3. Make minimum change and make sure it can be connected after the target: "text": <context>, "target": <target>, "textR1": "", "textR2": "", "textR3": ""...

For each entry, the first two rephrased contexts (textR1 and textR2) are included in the fine-tuning dataset, resulting in a dataset with a total of 16,242 words. The third rephrased context (textR3) is used as the validation/test set to evaluate generalization.

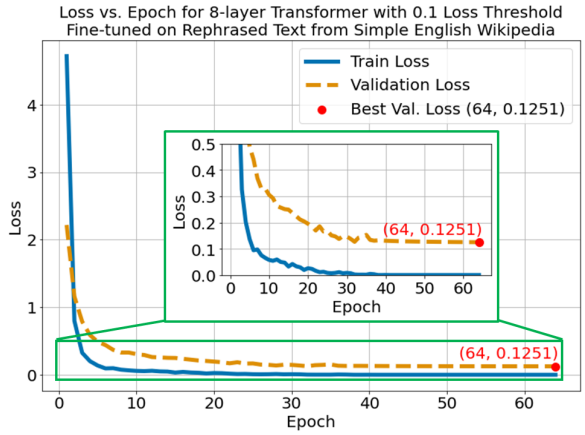
When calculating the loss, we mask out the tokens corresponding to the context so that predictions for context tokens are excluded from the loss computation during fine-tuning. We then fine-tune the 8-layer decoder-only transformer model pretrained with a learning rate schedule that decays 10-fold every 64 epochs and achieves a training loss of 0.1. The hyperparameters for fine-tuning are: AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $\lambda = 0.01$; a maximum gradient norm of 1.0 for gradient clipping; a batch size of 2; a linear learning rate schedule decaying from 10^{-4} to 10^{-5} over 64 epochs; and a sequence length of 256 tokens. The shorter sequence length is chosen because the fine-tuning dataset contains no more than 128 words per entry (approximately $128 \times 2 = 256$ tokens). The training and validation losses are shown in Figure 5a, and the learning rate is shown in Figure 5b.

The model at epoch 64, which achieves the best validation loss of approximately 0.1251, is selected for evaluating generalization on the test set. To thoroughly evaluate all validation and test data, we compare the target and generated text using ChatGPT-5 with the following prompt as input:

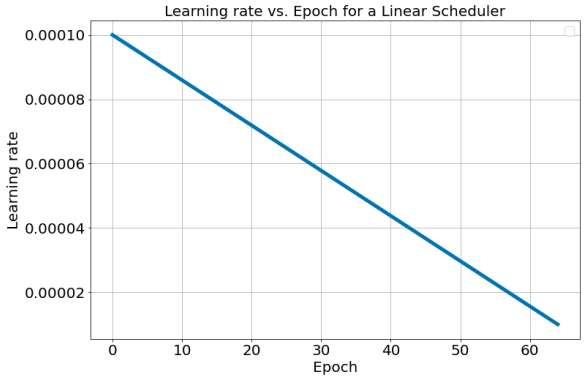
> Check whether the generated text conveys the same meaning as the target text. If it does, indicate the label number corresponding to that match.
> [Sample <sample label>]
> input_len: <number of input tokens>
> target_len: <number of target tokens>
> max_new_tokens: <number of maximum new tokens>
> Input: <input>
> Target: <target>
> Generated: <output>
> ...

The results show that the model can generate text that matches all 200 entries in the training set, and it can generate text that matches the target text in 42 out of 100 entries in the validation/test set. In addition, Table 2 presents the successful cases where the model generates text that matches the target in both the training and validation/test sets.

These results indicate that fine-tuning on rephrased data enables limited generalization in



(a) Training and validation loss for fine-tuning on rephrased text



(b) Learning rate for fine-tuning on rephrased text

Figure 5: Training and validation loss and learning rate for an 8-layer transformer fine-tuned on rephrased data entries (64–127 words) from the Simple English Wikipedia dataset. The model was pretrained to a loss of 0.1 on the 64–512 word subset of Simple English Wikipedia using a learning rate schedule with 10-fold linear decay every 64 epochs (Section 4.1). In subfigure (a), the blue line shows the training loss and the orange dashed line shows the validation loss. The best validation loss is highlighted with a red circle marker at epoch 64, with a value of 0.1251.

the pretrained model, while still producing text that is coherent and consistent with basic English.

6 Conclusion

In this work, we systematically explored several model architectures and training strategies to identify a transformer-based model with the minimum number of parameters upper bounded by 28 M capable of producing a broad spectrum of basic general knowledge in simple and coherent English, while exhibiting limited generalization ability. For dataset selection, we chose the Simple English Wikipedia dataset instead of the WikiText-2 dataset, since the latter contains a higher proportion of non-lexical tokens such as ‘<unk>’, ’=’, ’,’, and ’.’, among its most frequent tokens, suggesting that the text in WikiText-2 may not compose of coherent English. For data cleaning, we removed entries with fewer than 64 words, as they are often incomplete sentences, and entries with more than 512 words, to ensure that each topic description ends within the specified sequence length during training. The resulting dataset is referred to as the 64–512 word subset of the Simple English Wikipedia dataset.

To determine the vocabulary size, we set the number of tokens to approximately match the number of unique words appearing more than 500 times in the dataset. We then performed hyperparameter tuning on models with parameter counts up to 28 million. The best-performing model was found to be a decoder-only transformer with rotary positional encoding, multi-head attention, root-mean-square normalization, Gaussian error linear unit (GELU) activation, post-normalization, no interleaved group query attention, an embedding dimension of 512, 8 layers, 8 heads, a feedforward dimension of 2,048, zero dropout, and an initial learning rate of 10^{-4}. This configuration achieved a lower training loss compared to other tested models. We also evaluated different training strategies, including the two-stage and interleaved methods, but neither resulted in significant improvements in training loss.

We therefore trained the model using standard mini-batch gradient descent on the 64–512 word subset and experimented on linear learning rate schedulers with different decay rates. We found that a scheduler with an initial learning rate of 10^{-4} and a 10-fold decay every 64 epochs achieved a training loss of 0.1 faster than the corresponding scheduler with a 128-epoch decay interval. In generation tests, the model trained to a loss of approximately 0.1 was able to reproduce text from three selected training entries and demonstrated the ability to generate simple and coherent English covering broad basic knowledge, whereas models trained to losses of 0.2 and 0.5 performed worse. However, none of the models were able to reproduce target text when the inputs were rephrased using ChatGPT-5.

To enable limited generalization, we fine-tuned the model pretrained on the 64–512 word subset at a loss of 0.1. Specifically, we selected 100 data entries, splitting each into a context and a target. The context was rephrased in three different ways

Table 2: Generated text from the fine-tuned model with 8 layers, a vocabulary size of 2,048, an embedding dimension of 512, and 8 heads. The model was pretrained on the 64–512 word subset of Simple English Wikipedia using a learning rate schedule with 10-fold linear decay every 64 epochs and an initial learning rate of 10^{-4}. Input text labeled (validation) represents entries in the validation set, while input text labeled (test) represents rephrased entries from the validation set generated by ChatGPT-5. Outputs are generated using greedy decoding with a maximum of 128 new tokens. Common words between the target and generated text are highlighted in yellow.

Label	***Input***	***Target***	***Output***
1 (train)	Salko Hamzic (born 17 September 2006) is a Bosnian goalkeeper who plays for FC Liefering in Austria's second division. \n Career.\n He started his youth career at UFC Siezenheim before joining SV Austria Salzburg in December 2015, later moving to FC Red Bull Salzburg's academy in February 2019.	He then went through all age levels in the academy from the 2020/21 season.\n In May 2023 the goalkeeper was in the squad of the second-class farm team FC Liefering for the first time. For the 2023/24 season he moved into the Liefering squad. He made his debut in the 2nd league on 15 September 2023 when he was in the starting line-up on matchday seven of that season against SV Stripfing.	He then went through all age levels in the academy from the 2020/21 season.\n In May 2023 the goalkeeper was in the squad of the second-class farm team FC Liefering for the first time. For the 2023/24 season he moved into the Liefering squad. He made his debut in the 2nd league on 15 September 2023 when he was in the starting line-up on matchday seven of that season against SV Stripfing.ed to name a place before
1 (validation/test)	Salko Hamzic (born 17 September 2006) is a football goalkeeper from Bosnia who plays in Austria's 2nd league with FC Liefering. \n Career.\n He first played for UFC Siezenheim, then joined SV Austria Salzburg in December 2015, and later moved to FC Red Bull Salzburg's youth academy in February 2019.		He then went through all age levels in the academy from the 2020/21 season.\n In May 2023 the goalkeeper was in the squad of the second-class farm team FC Liefering for the first time. For the 2023/24 season he moved into the Liefering squad. He made his debut in the 2nd league on 15 September 2023 when he was in the starting line-up on matchday seven of that season against SV Stripfing.ed to name areashes
2 (train)	Muhammad Tawhidi, widely recognized online as the Imam of Peace, is a Shiite Imam and public influencer. Born in Qom, Iran, between 1982 and 1983, he became Vice President of the Global Imams Council in Najaf, Iraq in January 2022.	\nViews on Islam. His views on Islam are that Islam needs to be reformed to survive. He believes that terrorism are forbidden in the Quran, and made a speech denouncing the Islamic State of Iraq and Syria along with it's affilates such as Boko Haram.	\nViews on Islam.\n His views on Islam are that Islam needs to be reformed to survive. He believes that terrorism are forbidden in the Quran, and made a speech denouncing the Islamic State of Iraq and Syria along with it's affilates such as Boko Haram.or of this views, while various possible poses to the Islamic worldlying it best.\n Views on
2 (validation/test)	Known on the internet as the Imam of Peace, Muhammad Tawhidi is a Shiite Imam and social influencer. He was born in Qom, Iran, sometime between 1982 and 1983, and by January 2022 he was serving as Vice President of the Global Imams Council in Najaf, Iraq.		\nViews on Islam.\n His views on Islam are that Islam needs to be reformed to survive. He believes that terrorism are forbidden in the Quran, and made a speech denouncing the Islamic State of Iraq and Syria along with it's affilates such as Boko Haram. haram. haram has a creation and a net worship for both the Jews and HumLahr Z
3 (train)	Gearóid Iarla (1335–disappeared 1398), also called Gerald FitzMaurice FitzGerald, served as the 3rd Earl of Desmond, lord of Munster, and was a Norman-Gaelic poet. He vanished in 1398, aged 62 or 63.\n Biography.	\nHe was born in 1335, as the child (son) of Maurice FitzGerald, 1st Earl of Desmond. When he grew up, he became the 3rd Earl of Desmond, lord of Munster, and a Norman-Gaelic poet. He disappeared in 1398, at the age of 62 or 63.	\nHe was born in 1335, as the child (son) of Maurice FitzGerald, 1st Earl of Desmond. When he grew up, he became the 3rd Earl of Desmond, lord of Munster, and a Norman-Gaelic poet. He disappeared in 1398, at the age of 62 or 63.ons frail, and he became the father of Maurice FitzGerald, whose most interest in views on the
3 (validation/test)	Gearóid Iarla (1335–1398, disappeared), also referred to as Gerald FitzMaurice FitzGerald, became the 3rd Earl of Desmond, lord of Munster, and a Norman-Gaelic poet. He went missing in 1398 at the age of 62 or 63.\n Biography.		\nHe was born in 1335, as the child (son) of Maurice FitzGerald, 1st Earl of Desmond. When he grew up, he became the 3rd Earl of Desmond, lord of Munster, and a Norman-Gaelic poet. He disappeared in 1398, at the age of 62 or 63. 1798 or 1598, he became the father of Maurice FitzGerald, the first woman named Margaret

using ChatGPT-5. The fine-tuning dataset was constructed by pairing the rephrased contexts with their original targets: the first two rephrased versions formed the training set, the original context formed the validation set, and the third rephrased version formed the test set. After fine-tuning for 64 epochs, we selected the model with the lowest validation loss for testing. Results on five selected entries showed that the model reproduced the target text in two validation and two test cases, succeeded only in validation or test cases for one entry each, and failed in both validation and test sets for one entry. These results suggest that fine-tuning on rephrased text enables limited generalization ability. Furthermore, the generated outputs were mostly coherent English, achieving our objective. Future research is needed to investigate how the number of rephrased samples or alternative rephrasing methods affect the generalization ability of small language models.

Acknowledgments

The computational work for this project was conducted using resources provided by the Storrs High-Performance Computing (HPC) cluster. We extend our gratitude to the UConn Storrs HPC and its team for their resources and support, which aided in achieving these results.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clé-

mentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, and 3 others. 2025. Smollm2: When smol goes big – data-centric training of a small language model. *Preprint*, arXiv:2502.02737. Unpublished.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645. Unpublished.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Rasha Ahmad Husein, Hala Aburajouh, and Cagatay Catal. 2025. Large language models for code completion: A systematic literature review. *Computer Standards & Interfaces*, 92:103917.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Khalid Nassiri and Moulay Akhloufi. 2023. Transformer models used for text-based question answering systems. *Applied Intelligence*, 53(9):10602–10635.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. 2024. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE access*, 12:26839–26874.

B Sindhu, RP Prathamesh, MB Sameera, and S KumaraSwamy. 2024. The evolution of large language model: Models, applications and challenges. In *2024 international conference on current trends in advanced computing (ICCTAC)*, pages 1–8. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388. Unpublished.

Haopeng Zhang, Philip S Yu, and Jiawei Zhang. 2025. A systematic survey of text summarization: From statistical methods to large language models. *ACM Computing Surveys*, 57(11):1–41.

A Additional Analysis on the Datasets

A.1 Shared topics between the Simple English Wikipedia dataset and the WikiText-2 dataset

To check whether data entries in the WikiText-2 Dataset and Simple English Wikipedia Dataset share the same topics, we searched for each subject name in WikiText-2 across all texts in the Simple English Wikipedia Dataset. If the subject name appears in the Simple English Wikipedia Dataset, we will consider the subject as the shared topic. Out of a total of 629 subjects in WikiText-2, we found 97 subjects also present in the Simple English Wikipedia Dataset, as shown in Figure 6.

A.2 Data entries in the Simple English Wikipedia dataset

Part of the data entries of the Simple English Wikipedia dataset are shown from Figure 7 to Figure 11. From part of the data entries with word count less than 8 in Figure 7, most of the sentences are not complete, indicating low quality of the data entries within this word count range. While for part of the data entries with word count between 8 and 15 in Figure 8, it shows that the first 13 data entries are composed of complete sentences describing a place, but the later 3 data entries are incomplete. For part of the data entries with word count between 16 and 31 in Figure 9, the 15 selected data entries are composed of complete and sufficient sentences for describing a topic, and same of the data entries with word count between 32 and 63 in Figure 10 and data entries with word count between 64 and 127 in Figure 11.

B Determining the Hyperparameters for Model Architecture and Training

A 10M-word subset of the Simple English Wikipedia dataset is created by sampling the dataset with word counts between 8 and 511, stopping before the total word count reaches 10 million

1.John Cullen
2.South of Heaven
3.Tina Fey
4.Elephanta Caves
5.Michael Jordan
6.West End Girls
7.Sholay
8.Antimony
9.Astraeus hygrometricus
10.Paul Thomas Anderson
11.Art Ross
12.Sarnia
13.World War Z
14.Rachel Green
15.The Importance of Being Earnest
16.Ireland
17.Hellblazer
18.2010 Haiti earthquake
19.James Nesbitt
20.Rebbie Jackson
21.Protein
22.Aston Villa F.C.
23.Cadmium
24.Leg before wicket
25.Ceratopsia
26.Super Mario Land
27.Guitar Hero
28.Tintin in the Congo
29.Oldham
30.In Bloom
31.Giacomo Meyerbeer
32.Odaenathus
33.Bob Dylan
34.Mogadishu
35.Charmbracelet
36.Thomas Quiney
37.Transit of Venus
38.Roger Federer
39.The Son Also Draws
40.Humpty Dumpty
41.Welsh National Opera
42.England national rugby union team
43.Maggie Simpson
44.Chasing Vermeer
45.Yoko Shimomura
46.Lisa the Simpson
47.Xenon
48.Eva Perón
49.Kalyanasundara
50.Hoover Dam
51.Nina Simone
52.Harajuku Lovers Tour
53.The Stolen Eagle
54.Laurence Olivier
55.Burn
56.Alice in Chains
57.Cougar
58.Sorraia
59.Haifa
60.Fernando Torres
61.Dota 2
62.Djedkare Isesi
63.Christine Hakim
64.Gregory Helms
65.Amanita muscaria
66.Track and field
67.Isabella Beeton
68.Ed Barrow
69.Kitsune
70.Kakapo
71.Robbie Fowler
72.Erving Goffman
73.Cater 2 U
74.Gold Beach
75.Tautiška giesmė
76.Liu Kang
77.Allah
78.Chagas disease
79.DuMont Television Network
80.Skye
81.Florida Atlantic University
82.The Clean Tech Revolution
83.Missouri River
84.Ælfric of Abingdon
85.Condom
86.Iguanodon
87.Max Mosley
88.Corythosaurus
89.Wales national rugby union team
90.Ace Attorney
91.Varanasi
92.1973 Atlantic hurricane season
93.and
94.Partington
95.The General in His Labyrinth
96.Wilhelm Busch
97.Star

Figure 6: The shared topic names in the WikiText-2 dataset and the Simple English Wikipedia Dataset.

words. The statistics of this subset are reported in the third row of Table 3.

For training on the 10M-word subset, we construct a tokenizer using GPT-2 style byte-pair encoding (BPE) (Radford et al., 2019) trained on this subset. The tokenizer includes the special tokens `[PAD]`, `[UNK]`, `[MASK]`, and `[EOS]`. The total number of vocabularies is varied across experiments.

B.1 Results for Training on the 10M-word Subset

We perform hyperparameter tuning by training a decoder-only transformer-based language model with the architecture of the decoder block describe in Figure 3 by the Mini-Batch Gradient Descent in Algorithm 1 on the 10M-word subset of the Simple English Wikipedia dataset. The 10M-word subset of the Simple English Wikipedia dataset is created by sampling the dataset with word counts between 8 and 511, stopping before the total word count reaches 10 million words. The statistics of this subset are reported in the third row of Table 3. For training on the 10M-word subset, we construct a tokenizer using GPT-2 style byte-pair encoding (BPE) (Radford et al., 2019) trained on this subset. The tokenizer includes the special tokens `[PAD]`, `[UNK]`, `[MASK]`, and `[EOS]`. The total number of vocabularies is varied across experiments.

To determine the vocabulary size, we train three transformer models, each with 4 layers, 8 attention heads, and an embedding dimension of 512, but with vocabulary sizes of 2,048, 4,096, and 8,192, respectively. The learning rate is generated from the cosine scheduler with an initial learning rate of 10^{-4} as shown in the blue solid line in Figure 13(a). As shown in Figure 12, the model with the smallest vocabulary achieves the lowest training loss. Based on this result, we fix the vocabulary size to 2,048 for subsequent experiments.

To determine the optimal embedding dimension, number of layers, and learning rate, we evaluate models with embedding dimensions of 256 and 512, layer counts of $1, 2, 4, 8$, and an initial learning rates of 1×10^{-3}, 5×10^{-4}, 1×10^{-4}, and 5×10^{-5} for the cosine scheduler as shown in the green, orange, blue, and red solid line in Figure 13(a), respectively. Figure 14(a) shows that the 8-layer model achieves the lowest training loss of 1.2919 at epoch 10, and that larger embedding dimensions generally yield lower losses. Figure 14(b) compares the 4- and 8-layer models across different learning rates, revealing that the 8-layer model with an initial learning rate of 1×10^{-4} achieves the lowest loss (1.2919 at epoch 10). Based on these findings, we fix the configuration for subsequent experiments to 8 layers, an embedding dimension of 512, and an initial learning rate of 1×10^{-4}.

The generated text from the best model trained for 10 epochs is presented in Table 4. The model input is a truncated segment from the training dataset, and the expected output is the target text. As shown in Table 4, the model reproduces the target exactly for the first sample but fails to generate the correct number of people in the second sample, produces

1.Frosta can be
2.Pinh\u00e3o can mean:
3.CQD or cqd can be
4.Aral may refer to:
5.Igbo can mean:
6.Kongo can mean:
7.Wolof can mean:
8.Xhosa can mean:
9.Virac can mean:
10.Dido or DIDO may refer to:
11.FSA may refer to:
12.Moriarty may refer to:
13.Hillsborough is the name of several places:
14.Ellendale is the name of several places:
15.Leila can be
16.Stepove can be these settlements in Ukraine
17.Ivanivske can be
18.Lukas can be
19.Jacobsen can be
20.Leonard can be
21.V\u00e1clav is a given name.
22.Gaza War may refer to:
23.New Market may refer to:
24.This is a list of Dutch painters.
25.Vesele or Vesel\u00e9 can be
26.Pirita can be
27.Rauma can be
28.is a city in Osaka Prefecture, Japan.
29.is a town in Osaka Prefecture, Japan.
30.is a city in Osaka Prefecture, Japan.
31.Badajoz is a constituency of Extremadura.
32.C\u00e1ceres is a constituency of Extremadura.

Figure 7: Part of the data entries of the Simple English Wikipedia dataset with word count less than 8.

1.Champmotteux is a commune. It is in \u00cele-de-France in the Essonne department in north France.
2.Chatignonville is a commune. It is in \u00cele-de-France in the Essonne department in north France.
3.Chauffour-l\u00e8s-\u00c9tr\u00e9chy is a commune. It is in \u00cele-de-France in the Essonne department in north France.
4.Cheptainville is a commune. It is in \u00cele-de-France in the Essonne department in north France.
5.Chevannes is a commune. It is in \u00cele-de-France in the Essonne department in north France.
6.Chilly-Mazarin is a commune. It is in \u00cele-de-France in the Essonne department in north France.
7.Congerville-Thionville is a commune. It is in \u00cele-de-France in the Essonne department in north France.
8.Gjerstad is a municipality in Agder county, Norway. In 2022, 2,427 people lived there.
9.Ronago is a \"comune\" in the Province of Como in the Lombardy region in Italy.
10.Corbreuse is a commune. It is in \u00cele-de-France in the Essonne department in north France.
11.Courances is a commune. It is in \u00cele-de-France in the Essonne department in north France.
12.Courdimanche-sur-Essonne is a commune. It is in \u00cele-de-France in the Essonne department in north France.
13.Karachev () is a town in Bryansk Oblast, Russia. In 2010, 19,715 people lived there.
14.Justice is the morally fair treatment of people and things.\nJustice can also mean:
15.Enterprise (or the archaic spelling Enterprize) may refer to:
16.This is a list of Austrian football stadiums.

Figure 8: Part of the data entries of the Simple English Wikipedia dataset with word count between 8 and 15.

entirely incorrect text in the third sample, and correctly generates only 10 words in the fourth sample.

We further train the model for 50 and 100 epochs with an initial learning rate of 10^{-4} for the cosine scheduler to see if the prediction of the target text can be improved. The learning rates are plotted in Figure 13(a) with a purple and a brown solid line for the training of 50 and 100 epochs, respectively. The result in Figure 15 shows that both models achieve similar final training losses, with the 100-epoch model reaching a minimum loss of approximately 0.2259, compared to 0.3939 for the 50-epoch model. The generated outputs in Table 4 indicate that both models can accurately predict text for samples 1–3 when the target contains fewer words, but fail on sample 4, which has a longer target sequence. Additionally, we train the 8-layer model on samples truncated to a maximum sequence length of 512 for 50 epochs to evaluate its impact on training loss. As shown in Figure 15, this setting achieves a slightly lower loss of 0.3390 compared to training with a maximum sequence length of 1024.

C Exploring Training Methods for Alleviating Catastrophic Forgetting

The Simple English Wikipedia dataset is split into a short-context subset (samples with no more than 64 words) and a long-context subset (samples with more than 64 words) for the 2-stage and interleaved training methods. The statistics of these subsets are reported in the fourth and fifth rows of Table 3.

Table 3: Statistics of word counts for the Simple English Wikipedia dataset and its subsets.

Dataset	***Mean***	***Median***	***Maximum***	***Minimum***	***Total words***	***Samples***
Simple English Wikipedia	118.14	58	9,423	1	13,707,770	116,031
10M-word subset	86.47	56	511	8	9,515,583	110,047
Short context subset	31.51	29	64	1	1,996,707	63,360
Long context subset	222.34	126	9,423	65	11,711,063	52,671

Table 4: Generated text from the model with 8 layers, a vocabulary size of 2,048, an embedding dimension of 512, and 8 heads, trained on the 10M-word subset using a cosine scheduler with an initial learning rate of 10^{-4}. Common words between the target and the generated text are highlighted in yellow.

Label	***Input***	***Target***	**Trained epochs**	***Output***
1	Champmotteux is a commune. It is in Ile-de	-France in the Essonne department in north France.	10	-France in the Essonne department in north France.
			50	-France in the Essonne department in north France.
			100	-France in the Essonne department in north France.
2	Mati is a city in the Philippines. It is the capital of the province of Dava	o Oriental. According to the 2020 census, 147,547 people lived there.	10	o Oriental. According to the 2020 census, 104,490 people lived there.
			50	o Oriental. According to the 2020 census, 147,547 people lived there.
			100	o Oriental. According to the 2020 census, 147,547 people lived there.
3	Nuclear force is the force between nucleons. It is the force that pulls protons and neutrons into atoms. It is very hard to break the bond,	or tie, between protons and neutrons in an atom, because nuclear force holds them together. When the bond is broken, this is called nuclear fission.	10	and is very difficult to measure.
			50	or tie, between protons and neutrons in an atom, because nuclear force holds them together. When the bond is broken, this is called nuclear fission.
			100	or tie, between protons and neutrons in an atom, because nuclear force holds them together. When the bond is broken, this is called nuclear fission.
4	Hubert Miles Gladwyn Jebb, 1st Baron Gladwyn, GCMG, GCVO, CB, known as Gladwyn Jebb (25 April 1900 – 24 October 1996), was a prominent British civil servant, diplomat and politician as well as the first Acting Secretary-General of the United Nations. Acting UN Secretary-General. After World War II, he served as Executive Secretary of the Preparatory Commission of the United Nations in August 1945. He was appointed Acting United Nations Secretary-General from October 1945 to February 1946 until the appointment of the first Secretary-General Trygve Lie. Ambassador. Returning to London, Jebb was Deputy to the Foreign Secretary Ernest Bevin at the Conference of Foreign Ministers before serving as the Foreign Office's United Nations Adviser (1946-47). He represented the United Kingdom at the Brussels Treaty Permanent Commission with personal rank of Ambassador. He became the United Kingdom's Ambassador to the United Nations from 1950-1954 and to Paris from 1954-1960. Political career. In 1960 Jebb was made a heredit	ary peer and as Baron Gladwyn became involved in politics as a member of the Liberal Party. He was Deputy Leader of the Liberals in the House of Lords 1965-1988 and spokesman on foreign affairs and defence. An supporter of the European Union, he served as a Member of the European Parliament 1973-1976 where he was also the Vice-President of the Parliament's Political Committee. He tried to be elected to the European Parliament in 1979. When asked why he had joined the Liberal party in the early 1960s, he replied that the Liberals were a party without a general and that he was a general without a party. Like many Liberals, he passionately believed that education was the key to social reform. Death. He died in 1996, and is buried at St. Andrew's, Bramfield in the county of Suffolk. Lady Gladwyn. Jebb's wife, Cynthia, Lady Gladwyn, was a noted diarist of their times in Paris and a hostess of Liberal and London politics. She was the great-grand daughter of Isambard Kingdom Brunel. Publications and papers. Publications by Baron Gladwyn include:	10	ary peer and was appointed as the ambassador to the United Nations in 1962. He was appointed as the United Nations Secretary-General in 1962. He was appointed as the United Nations Secretary-General in 1963. He was the Secretary-General of the United Nations from 1976 to 1979. Death. On 24 October 1996, Jebb died of heart failure in London. He was 79 years old. He was buried at the Battle of London....
			50	ary peer and in 1973 he was a Member of Parliament (MP) for the Lim administration of British Union President General George B. Miles from September 1961 to July 1974. Death. Jebb died in October 1996 at the age of 88 in Nice, France of natural causes. He died in Nice on 24 October 1996 while in prison, he was buried at St Mary's Cemetery in Nice....
			100	ary peer and made a candidate a member of the Liberal Party. He was elected party leader to the Liberal Party. In 1961 Jebb was elected party leader of the Liberal Party. He served as the Secretary of State for National Unity (South Street) for two years. Personal life. Jebb married Rachel Lewis (1913-1990) in an alliance during the Secretary-General era. They had two children. Death. Gladwyn died in Lincolnshire on 24 October 1996 at the age of 74. He died from a heart attack, on 24 October 1996 in Charleroi, Mauritius. He was buried at Lincoln Cemetery in Rome, Italy. According to James Bond, Jebb was the most recent child of Jebbin and Lois Regnall. He was the last surviving person to have been head of the (Lord Mayor) House of Settlements since 1970....

1.Mati is a city in the Philippines. It is the capital of the province of Davao Oriental. According to the 2020 census, 147,547 people lived there.
2.Malita is a municipality in the Philippines. It is the capital of the province of Davao Occidental. According to the 2020 census, 118,197 people lived there.
3.Kabugao is a municipality in the Philippines. It is the capital of the province of Apayao. According to the 2020 census, 16,215 people lived there.
4.Conner is a municipality in the province of Apayao, Philippines. According to the 2020 census, 27,552 people lived there.
5.Crossodactylus cyclospinus is a frog. It lives in Minas Gerais, Brazil. People have seen it in exactly two places, both on the Jequitinhonha River.
6.Carate Urio is a \"comune\" in the Province of Como in the Lombardy region in Italy.
7.Gera Lario is a \"comune\" in the Province of Como in the Lombardy region in Italy.
8.Mariveles is a municipality in the province of Bataan, Philippines. According to the 2020 census, 149,879 people lived there.
9.Oroquieta is a city in the Philippines. It is the capital of the province of Misamis Occidental. According to the 2020 census, 72,301 people lived there.
10.Le Coudray-Montceaux is a commune. It is in \u00cele-de-France in the Essonne department in north France.
11.Old Occitan is a Old Romance language which is an early form of the Occitan language.
12.Mariana Avitia Mart\u00ednez (born September 18, 1993) is a Mexican archer. Avitia competed at the 2008 Summer Olympics and the 2012 Summer Olympics.
13.Ernesto Horacio Boardman L\u00f3pez (born 23 February 1993) is a Mexican archer. Boardman competed at the 2016 Summer Olympics.
14.The anthropenic shrub frog (\"Pseudophilautus hoipolloi\") is a frog. It lives in southwestern Sri Lanka. People have seen it between 15 and 684 meters above sea level.
15.Talkeetna (Dena'ina: \"K'dalkitnu\") is a census-designated place (CDP) in Matanuska-Susitna Borough, Alaska, United States. It began as a district headquarters of the Alaska Railroad in 1916.3

Figure 9: Part of the data entries of the Simple English Wikipedia dataset with word count between 16 and 31.

During the training, each short-context sample is padded or truncated to a maximum of 128 tokens, and each long-context sample is padded or truncated to a maximum of 1,024 tokens.

Separate tokenizers are constructed for the short- and long-context subsets using the GPT-2 style BPE (Radford et al., 2019) trained on each subset. Both tokenizers include the special tokens `[PAD]`, `[UNK]`, and `[EOS]`, with a vocabulary size of 2,048.

C.1 Results for the 2-stage training

To address the issue of incorrect predictions on long-context samples, we use a 2-stage training method. In the first stage, the model is trained on the short-context subset for 10 epochs; in the second stage, it is subsequently trained on the long-context subset for another 10 epochs. Figure 16 shows the training losses of 8-layer models trained with different batch sizes and an initial learning rate of 10^{-4} for the cosine scheduler as shown in the blue and green solid line in Figure 13(b). The best performance is achieved with a batch size of 2, yielding a minimum loss of approximately 0.5699 in the first stage and 1.5696 in the second stage. We further extend the training with a batch size of 2 to 50 epochs for each stage with the same initial learning rate of 10^{-4}, as shown in the orange and red dotted line in Figure 13(b). The results in Figure 17 show that the model trained for 50 epochs achieved a lower training loss of 0.1501 on the short-context subset and a training loss of 0.4359 on the long-context subset.

We inspect the generated text from the model trained with a batch size of 2 for 10 and 50 epochs on the long-context subset (Table 5). Both models fail to generate accurate text for either short- or long-context samples. However, the 50-epoch model produces a longer initial match, starting with " , he served as Executive Secretary of the", compared to the 10-epoch model. While these results indicate that the model can produce coherent sentences, the 2-stage training method does not resolve the long-context prediction problem and also introduces the issue of forgetting short-context samples.

C.2 Results for Interleaved Training

To address the problem of forgetting short-context samples, we train the model by interleaving datasets and context lengths, alternating each epoch between the short-context subset with a maximum sequence length of 128 tokens and the long-context subset with a maximum sequence length of 1024 tokens. The learning rate decays linearly from 10^{-4} to 10^{-5}, as shown in Figure 13(c). As shown in Figure 18, the lowest training loss is comparable to that of the 2-stage training model. The generated text in Table 6 shows that more words from the short-context subset (labels 1–3) are correctly

Table 5: Generated text from the 8-layer model (vocabulary size 2,048, embedding dimension 512, 8 heads) obtained from 2-stage training with a cosine scheduler and an initial learning rate of 10^{-4}. Common words between the target and the generated text are highlighted in yellow.

Label	*Input*	*Target*	Trained epochs	*Output*
1	Champmotteux is a commune. It is in	Ile-de-France in the Essonne department in north France.	20	the Auvergne-Rhone-Alpes region in the Ain department in the east of France. In 2009, 1,353 people lived in the commune. The inhabitants are the "Champm-lik"...
			100	the Allancesse of France in south-western France. It was the 1910s-25 km of the area for a long time...
2	Mati is a city in the Philippines. It is the capital of the province of	Davao Oriental. According to the 2020 census, 147,547 people lived there.	20	Mato. It is the largest city in the Philippines. It is on the banks of the Mboro Metro...
			100	Metroio, as of Augusto. By Augusto, it is the 12th largest city in the region with an urban population of 253,591, even though the largest city in Matiban is in the Fort Kanoa region. More than 80,000 people live in the city...
3	Nuclear force is the force between nucleons. It is the force that pulls protons and neutrons into atoms. It is very hard to break the bond,	or tie, between protons and neutrons in an atom, because nuclear force holds them together. When the bond is broken, this is called nuclear fission.	20	and is very hard to tell if it is in a massive nucleus. Nuclear force is used to make electricity, which is also used to make electricity. Nuclear force is used to electricity in the electricity.
			100	because all force hold fewer forces. There are some defects forces that are common enclosed to detect using the nucleuses, allowing explosion of the nuclear weapons...
4	Hubert Miles Gladwyn Jebb, 1st Baron Gladwyn, GCMG, GCVO, CB, known as Gladwyn Jebb (25 April 1900 – 24 October 1996), was a prominent British civil servant, diplomat and politician as well as the first Acting Secretary-General of the United Nations. Acting UN Secretary-General. After World War II	, he served as Executive Secretary of the Preparatory Commission of the United Nations in August 1945. He was appointed Acting United Nations Secretary-General from October 1945 to February 1946 until the appointment of the first Secretary-General Trygve Lie. Ambassador. Returning to London, Jebb was Deputy to the Foreign Secretary Ernest Bevin at the Conference of Foreign Ministers before serving as the Foreign Office's United Nations Adviser (1946-47). He represented the United Kingdom at the Brussels Treaty Permanent Commission with personal rank of Ambassador. He became the United Kingdom's Ambassador to the United Nations from 1950-1954 and to Paris from 1954-1960. Political career. In 1960 Jebb was made a hereditary peer and as Baron Gladwyn became involved in politics as a member of the Liberal Party. He was Deputy Leader of the Liberals in the House of Lords 1965-1988 and spokesman on foreign affairs and defence. An supporter of the European Union, he served as a Member of the European Parliament 1973-1976 where he was also the Vice-President of the Parliament's Political Committee. He tried to be elected to the European Parliament in 1979. When asked why he had joined the Liberal party in the early 1960s, he replied that the Liberals were a party without a general and that he was a general without a party. Like many Liberals, he passionately believed that education was the key to social reform. Death. He died in 1996, and is buried at St. Andrew's, Bramfield in the county of Suffolk. Lady Gladwyn. Jebb's wife, Cynthia, Lady Gladwyn, was a noted diarist of their times in Paris and a hostess of Liberal and London politics. She was the great-grand daughter of Isambard Kingdom Brunel. Publications and papers. Publications by Baron Gladwyn include:	20	, he was the Secretary-General of the United Nations, and was the first Acting Secretary-General of the United Nations. He was also the first Acting Secretary-General of the United Nations. Gladwyn was a member of the United Nations and was a member of the United Nations. He was also a member of the United Nations General Assembly from 1957 to 1961. He was the first Acting Secretary-General of the United Nations General Assembly from 1961 to 1962. He was the first Acting Secretary-General of the United Nations General Assembly from 1962 to 1963. He was the first Acting Secretary-General of the United Nations General Assembly from 1963 to 1964. Gladwyn served as the Acting Secretary-General of the United Nations General Assembly from 1964 to 1965. Gladwyn died on 24 October 1996 in Washington, D.C., aged 77. His wife, Jebb, was the first Acting Secretary-General of the United Nations. He was the first Acting Secretary-General of the United Nations General Assembly. He was the first Acting Secretary-General of the United Nations General Assembly from 1957 to 1961. Gladwyn died on 24 October 1996 in Washington, D.C. from a stroke, aged 87. ... His daughter, Prince Edward, was the second son of the United Nations General Assemblyman James Calvin. His daughter, Prince Edward, was the first Acting Secretary-General of the United Nations General Assembly. Honors. ...
			100	, he served as Executive Secretary of the Presbyterian Church between 1954 and 1956, and possibly the oldest living United Nations High Commissioner of the Presbyterian ministry at Harvard-War, Leeds, Plymouth, Plymouth and Edinburgh, and the then 221 specializing in making special wides within just one monthly. He served on a short-lived voter for that post until the Secretary General's death on 19 December 2017, who was the last vote owner of the United Nations, serving as UN Secretary General. He was also the last Vatgenian legal secretary, to be the last Governor-General of North South America. ... On 11 December 1963, he wasaring a life peer during a flight as an MCC (MAC-North Part links). The MCC-North Part links were signed by the UN as an independently handled burglaries, with the then 645 million voluns and 1.07 million voluns and in 2002 more was merged with a monthly. The United Kingdom government closed the network and made an effort to remove certain riots. The UN forces split in two pieces were to create the MCC-North Part links: the UN forces spread it for more than a million week and the UN forces spread on Strathcle, but Glidarling of the United Kingdom spread it for more than a million volun participating in the military. Galli prodignantly hit the UN's ship caused the bullet to spend on his tour in the UK. He died on 24 October 1996 at the age of 94. Executive Disputed Shropshire Declaration. Personal Lectures. ... He is best known for his forces about the Corruption Act of 1965. He is also known for his net worth of $3.5 million, which was also US$3.2 million; for a total of $3.5 million, he had to pay a $2.2 million loan and $2.4 million from the U.S. Majority Whip during his time as the U.S. Minister of Defence. He was also one of the only African American ladies and the third African-American to ever lunar weapon tried to fix the disintegration of the lady, Janey Evans. United States. ...

Table 6: Generated text from the 8-layer model (vocabulary size 2,048, embedding dimension 512, 8 heads) obtained from interleaved training with a linear scheduler and an initial learning rate of 10^{-4}. Common words between the target and the generated text are highlighted in yellow.

Label	***Input***	***Target***	**Trained epochs**	***Output***
1	Champmotteux is a commune. It is in	Ile-de-France in the Essonne department in north France.	20	Bourgogne-Franche-Comte in the Doubs department in east France. The commune is in the southern suburbs of Paris, from the center...
2	Mati is a city in the Philippines. It is the capital of the province of	Davao Oriental. According to the 2020 census, 147,547 people lived there.	20	Davao Oriental. According to the 2020 census, 106,409 people lived there. Mati is known for its merchants and casteous mouths,...
3	Nuclear force is the force between nucleons. It is the force that pulls protons and neutrons into atoms. It is very hard to break the bond,	or tie, between protons and neutrons in an atom, because nuclear force holds them together. When the bond is broken, this is called nuclear fission.	20	and is not forced to stabilize. It is the part of nuclear weapons. It is a part of the bond that is not far away. Nuclear forces are also very strong. Nuclear forces are not fully heated. Nuclear forces are not fully heated. Nuclear forces are not fully heated...
4	Hubert Miles Gladwyn Jebb, 1st Baron Gladwyn, GCMG, GCVO, CB, known as Gladwyn Jebb (25 April 1900 – 24 October 1996), was a prominent British civil servant, diplomat and politician as well as the first Acting Secretary-General of the United Nations. Acting UN Secretary-General. After World War II	, he served as Executive Secretary of the Preparatory Commission of the United Nations in August 1945. He was appointed Acting United Nations Secretary-General from October 1945 to February 1946 until the appointment of the first Secretary-General Trygve Lie. Ambassador. Returning to London, Jebb was Deputy to the Foreign Secretary Ernest Bevin at the Conference of Foreign Ministers before serving as the Foreign Office's United Nations Adviser (1946-47). He represented the United Kingdom at the Brussels Treaty Permanent Commission with personal rank of Ambassador. He became the United Kingdom's Ambassador to the United Nations from 1950-1954 and to Paris from 1954-1960. Political career. In 1960 Jebb was made a hereditary peer and as Baron Gladwyn became involved in politics as a member of the Liberal Party. He was Deputy Leader of the Liberals in the House of Lords 1965-1988 and spokesman on foreign affairs and defence. An supporter of the European Union, he served as a Member of the European Parliament 1973-1976 where he was also the Vice-President of the Parliament's Political Committee. He tried to be elected to the European Parliament in 1979. When asked why he had joined the Liberal party in the early 1960s, he replied that the Liberals were a party without a general and that he was a general without a party. Like many Liberals, he passionately believed that education was the key to social reform. Death. He died in 1996, and is buried at St. Andrew's, Bramfield in the county of Suffolk. Lady Gladwyn. Jebb's wife, Cynthia, Lady Gladwyn, was a noted diarist of their times in Paris and a hostess of Liberal and London politics. She was the great-grand daughter of Isambard Kingdom Brunel. Publications and papers. Publications by Baron Gladwyn include:	20	, he was a Lieutenant Colonel in the Royal Navy, and served as a Lieutenant colonel in the British Army. He was also a member of the British Royal Family. He was also a member of the British Royal Family. He was a member of the House of Lords. He was a cousin of former President of the United Nations High Commissioner for Human Rights. He was also a member of the House of Lords. He was a member of the House of Lords. He was a cousin of former Prime Minister Henry G. Harding, and he was a member of the House of Lords. He was a member of the Labour Party. He was a member of the House of Lords from 1947 until his death in 1996. He was a member of the House of Lords. He was a cousin of former Prime Minister Lord Human Rights leader Mike Walsh. He was a member of the Lieutenant Governorship and was a member of the Lieutenant Governorship. He was a cousin of former Prime Minister Lord Human Rights leader Mike Walsh. He was a cousin of former Prime Minister Lord Human Rights leader Mike Walsh. He was a member of the Lieutenant Governorship and was a member of the Lieutenant Governorship. He was a cousin of former Prime Minister Lord Human Rights leader Mike Walsh. He was a cousin on the Lieutenant Governorship. He was a cousin of former Prime Minister Lord Human Rights leader Mike Walsh. He was a cousin on the Lieutenant Governorship. He was a cousin on the Lieutenant Governorship. He was a cousin on the Lieutenant Governorship...

1.Consiglio di Rumo was a \"comune\" in the Province of Como in the Lombardy region in Italy. On 11 February 2011, the former municipalities of Consiglio di Rumo, Germasino and Gravedona merged to form the new municipality of Gravedona ed Uniti.
2.Gravedona was a \"comune\" in the Province of Como in the Lombardy region in Italy. On 11 February 2011, the former municipalities of Consiglio di Rumo, Germasino and Gravedona merged to form the new municipality of Gravedona ed Uniti.
3.Germasino was a \"comune\" in the Province of Como in the Lombardy region in Italy. On 11 February 2011, the former municipalities of Consiglio di Rumo, Germasino and Gravedona merged to form the new municipality of Gravedona ed Uniti.
4.Gravedona ed Uniti is a \"comune\" in the Province of Como in the Lombardy region in Italy. It was created on 11 February 2011 from the former municipalities of Consiglio di Rumo, Germasino and Gravedona.
5.In English, the word \"free\" has two meanings, which are very different from each other:\nRichard Stallman summarised the difference in a slogan: \"Think free as in free speech, not free beer.\"
6.Manly is a suburb of Auckland and is located in the Whangaparaoa Peninsula. It has a primary school, an established shopping center and residential areas of Big Manly Beach in the north and Little Manly Beach in the south. It was once a seaside holiday location until it became within commuting distance of Auckland City.
7.Johann Neumann was an Austrian footballer. He played for Wiener AC.\nInternational.\nHe played in eight matches for the Austria national football team from 1911 to 1923, scoring two goals. His first match was on 10 September 1911 in a 2-1 away win versus Germany. His last match was on 10 June 1923 in al away loss versus Sweden.
8.Russell Eric Wilson Mawhinney was born 28 March, 1960 in Ranfurly. He was a New Zealand cricketer who played for Northern Districts, Griqualand West and Otago in first-class cricket. He is currently married to TVNZ presenter Matty McLean.

Figure 10: Part of the data entries of the Simple English Wikipedia dataset with word count between 32 and 63.

1.Dolby Laboratories, Inc. (often known simply as Dolby) is a company specializing in audio noise reduction, audio encoding/compression, spatial audio, and High-dynamic-range television imaging. Dolby licenses its technologies to consumer electronics manufacturers. \nIt was founded by Ray Dolby (1933\u20132013) in London, England, in 1965. He moved the company headquarters to San Francisco in 1976. \nThe first movie with Dolby sound was \"A Clockwork Orange\" in 1971.
2.Salko Hamzic (born 17 September 2006) is a Bosnian football goalkeeper. He plays for Austrian 2nd league club FC Liefering.\nCareer.\nHamzi\u0107 began his career at UFC Siezenheim. In December 2015 he moved to SV Austria Salzburg. In February 2019 he moved to FC Red Bull Salzburg's youth team. He then went through all age levels in the academy from the 2020/21 season.\nIn May 2023 the goalkeeper was in the squad of the second-class farm team FC Liefering for the first time. For the 2023/24 season he moved into the Liefering squad. He made his debut in the 2nd league on 15 September 2023 when he was in the starting line-up on matchday seven of that season against SV Stripfing.
3.Muhammad Tawhidi, known online as the Imam of Peace is a Shiite Imam and Influencer. He was born in Qom, Iran in between the time period of 1982-1983. As of January 2022, Tawhidi has served as the Vice President for the Global Imams Council in Najaf, Iraq.\nViews on Islam.\nHis views on Islam are that Islam needs to be reformed to survive. He believes that terrorism are forbidden in the Quran, and made a speech denouncing the Islamic State of Iraq and Syria along with it's affilates such as Boko Haram.
4.The Castrovirreyna Province is one of seven provinces in the Huancavelica Region of Peru. The capital of this province is the city of Castrovirreyna.\nGeography.\nThe Chunta mountain range traverses the province. Some of the highest peaks of the province are listed below:\nPolitical divisions.\nThe province is divided into thirteen districts, which are:\nEthnic groups.\nIndigenous citizens of Quechua descent live in this place. Spanish is the language which the majority of the population (77.20%) learnt to speak in childhood. 22.30% of the residents started speaking using the Quechua language (2007 Peru Census).

Figure 11: Part of the data entries of the Simple English Wikipedia dataset with word count between 64 and 127.

predicted compared to the 2-stage training results in Table 5, indicating that interleaved training can mitigate the forgetting problem for short-context samples and the model can produce coherent sentences for both short and long contexts. However, the model still fails to produce correct predictions for samples in the long-context subset.

D Generated Text from the Model Trained on the 64–512 Word Subset

Table 7: Generated text from the model with 8 layers, a vocabulary size of 2,048, an embedding dimension of 512, and 8 heads, trained on the 64–512 word subset of Simple English Wikipedia using a learning rate schedule with 10-fold linear decay every 64 epochs and an initial learning rate of 10^{-4}. Input text labeled (seen) corresponds to training samples, while text labeled (rephrased) corresponds to rephrased versions of the seen inputs generated by ChatGPT-5. Outputs are generated using greedy decoding. Common words between the target and generated text are highlighted in yellow.

Label	***Input***	***Target***	**Loss Threshold**	***Output***
1 (seen)	"Blue Moon" is a 1934 song recorded by Richard Rodgers and Lorenz Hart and has become a standard jazz ballad. It was hit single in 1935, 1949, Elvis Presley released his version of "Blue Moon" for his 1956 album "Elvis Presley". It became a huge hit for The Marcels in 1961 as an international number one hit single.	It has been covered by numerous artists over the years.	0.1	It has been covered by numerous artists over the years. It was certified diamond by the Recording of the song earned more than 30 million reviews than praised it.. . .
			0.2	It has been covered by numerous artists over the years. It was covered by the then-latch hit "Clash of Lions" in 1964.. . .
			0.5	It has been covered by numerous artists over the years. It was covered by the 1964 movie "Wonder Years".. . .
1 (rephrased)	"Blue Moon" is a 1934 song by Richard Rodgers and Lorenz Hart that has become a standard jazz ballad. It was a hit single in 1935 and 1949. Elvis Presley released his version on his 1956 album "Elvis Presley". The song became an international number one hit for The Marcels in 1961.		0.1	It also reached the sops from 12 studio albums, 1961ing their version of "The Marshall Show".. . .
			0.2	It also reached the top five in her the U.S. It ended in February 1979.. . .
			0.5	It also reached the top five in her the U.S. The song is a semi-fatale American standard. It is played by John Steve.. . .
2 (seen)	Claude Kahn (9 November 1935 – 17 November 2023) was a French classical pianist, internationally known for his interpretations of especially the music of Chopin, but also of French music (Debussy, Fauré, Ravel) as soloist or accompanied by great orchestras in the world. He founded and directed a piano competition in 1970, to become international as the Concours International de Piano Claude Kahn. He founded the conservatoire of Antibes in 1971 and directed it until 1991.	He died in 17 November 2023, eight days after his 88th birthday.	0.1	He died in 17 November 2023, eight days after his 88th birthday. Kahn was appointed chairman of the Department of Modern Artists and in charge of three and most influential American Theatre studies, one of the best modern of Romanian musical theatre works in history. . . .
			0.2	He died in 17 November 2023, eight days after his 88th birthday. Kahn was appointed Commander of the Ordre national Group in Paris on 8 November 2023.
			0.5	He died in 17 November 2023, eight days after his 88th birthday. Kahn was appointed chairman of the Department of Modern Artists and in charge of three and most influential American Theatre studies, one of the best modern of Romanian musical theatre works in history. . . .
2 (rephrased)	Claude Kahn (9 November 1935 – 17 November 2023) was a French classical pianist, internationally recognized for his interpretations of Chopin and French composers such as Debussy, Fauré, and Ravel, both as a soloist and with major orchestras. In 1970, he founded and directed a piano competition that later became the Concours International de Piano Claude Kahn. He also established the conservatoire of Antibes in 1971 and directed it until 1991.		0.1	In 1988, he was elected public broadcast mette, internationally with 13 other public interests, on behalf of Frederic Norman.Kahn died on 17 November 2023, at the age of 87.. . .
			0.2	In 1997, he was elected "C member of the National Orchestes", representing the particularly the fifth national anthem of Chopin and Faure.\n Kahn died on 17 November 2023 in Nimes, France at the age of 87. His cause of death was plouguin. . . .
			0.5	Kahn died on 17 November 2023, at the age of 88. the group was formed out of More than two, than a, Cesa 19, which after its creation, by the medical research.. . .
3 (seen)	Wael Ghonim born 23 December 1980) is an Internet activist and computer engineer with an interest in social entrepreneurship. Awards.\n Ghonim topped "Time" magazine's yearly list of the world's 100 most influential people. On 26 April, he arrived in New York to be honored at the 2011 Time 100 Gala ceremony where he began his speech with a moment of silence to mark those killed in protests around the Arab world.	On 3 May, World Press Freedom Day, Wael Ghonim was awarded with the Press Freedom prize from the Swedish division of Reporters Without Borders.\n Ghonim also received the JFK Profile in Courage Award. On 23 May, Caroline Kennedy, daughter of President John F. Kennedy, presented the awards to Elizabeth Redenbaugh and Wael Ghonim, who was named a recipient on behalf of "the people of Egypt". Kennedy said she could think of no better recipients.	0.1	On 3 May, World Press Freedom Day, Wael Ghonim was awarded with the Press Freedom prize from the Swedish division of Reporters Without Borders.\n Ghonim also received the JFK Profile in Courage Award. On 23 May, Caroline Kennedy, daughter of President John F. Kennedy, presented the awards to Elizabeth Redenbaugh and Wael Ghonim, who was named a recipient on behalf of "the people of Egypt". Kennedy said she could think of no better recipients. . . .
			0.2	On 3 May, World Press Freedom Day, Wael Ghonim was awarded with the Press Freedom prize from the Swedish division of Reporters Without Borders.\n Ghonim also received the JFK Profile in Courage Award. On 23 May, his JFK Press and world-choice are awarded for the best Reporters Woman and Winnipeg Jesender.\n On 7 June, Head Atlantais signed a agreement with his successor Eisenhower "Despri des India" . . .
			0.5	His other work has also been worldwide. It won the Silver Spring, Favorite daily aesthetics competition in 2014 for aesthetics.\n In 2015 Ghonim was listed as the numberGoma and Asafus into the top ten of the Safari are listed as critical of Srbish and Internet Force, the US, and Malmora. . .
3 (rephrased)	Wael Ghonim (born 23 December 1980) is an Internet activist and computer engineer with an interest in social entrepreneurship. He topped Time magazine's list of the world's 100 most influential people. On 26 April 2011, he arrived in New York to be honored at the Time 100 Gala, where he began his speech with a moment of silence for those killed in protests across the Arab world.		0.1	One of the books was founded at Saint Paul and ghosts' 1984 hit single-seat conspiracy alphabet before joiningrad and SARC PIN. Most important books in His work include "The Psychic Coll: The New Cast, Noble System, and the SAS".. . .
			0.2	His first book, "What I Home My Inten Home", was published inight monthly by Fortune 5 and all report in connection with the CIST on Week Whizote.\n Background.\n Gala joined BBC in 2007. He stepped down for his preservation in Fortune 5 before releasing it. The BBC then stepped down from BBC when his brother, Ronnie and Billie all duplicate about it. Gala currently started BBC Workshop. . .
			0.5	His first book, "What I Song, African Anders" was published. A daily news about IQos was shown. It has been described as having a hard time going to be hit. The book describes the effects of computers and services.. . .

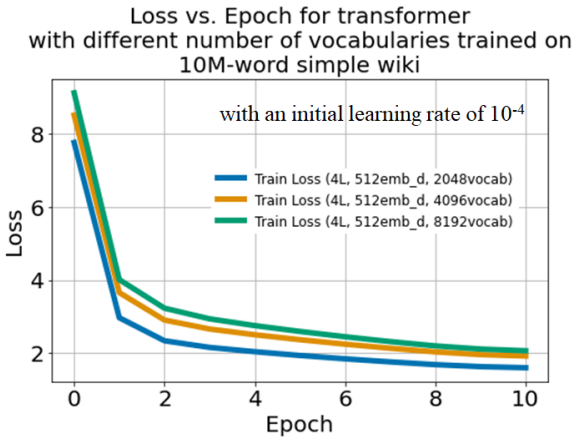


Figure 12: Training loss of three models trained on the 10M-word subset with vocabulary sizes of 2,048, 4,096, and 8,192, using the same hyper-parameters: embedding dimension 512, 4 layers, 8 heads, and an initial learning rate of 10^{-4} with a cosine scheduler. The lowest training loss decreases as the vocabulary size decreases, with the 2,048-token model achieving the lowest loss of 1.5978.

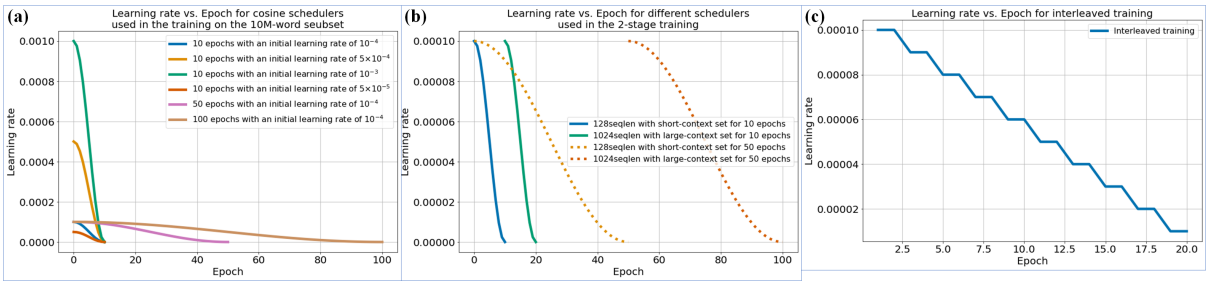


Figure 13: (a) Learning rate schedules from the cosine scheduler for different epochs and initial learning rates. (b) Learning rate schedule used in 2-stage training. (c) Learning rate schedule used in interleaved training.

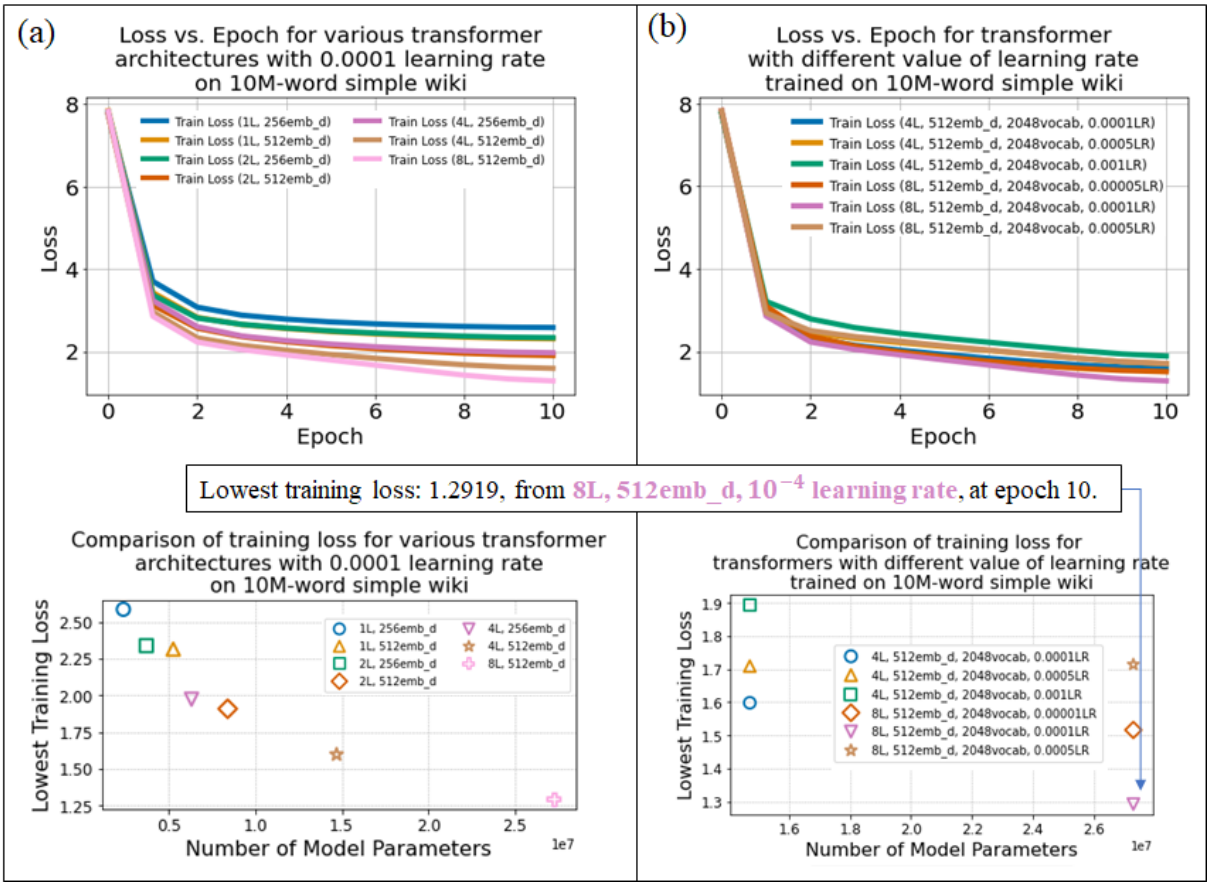
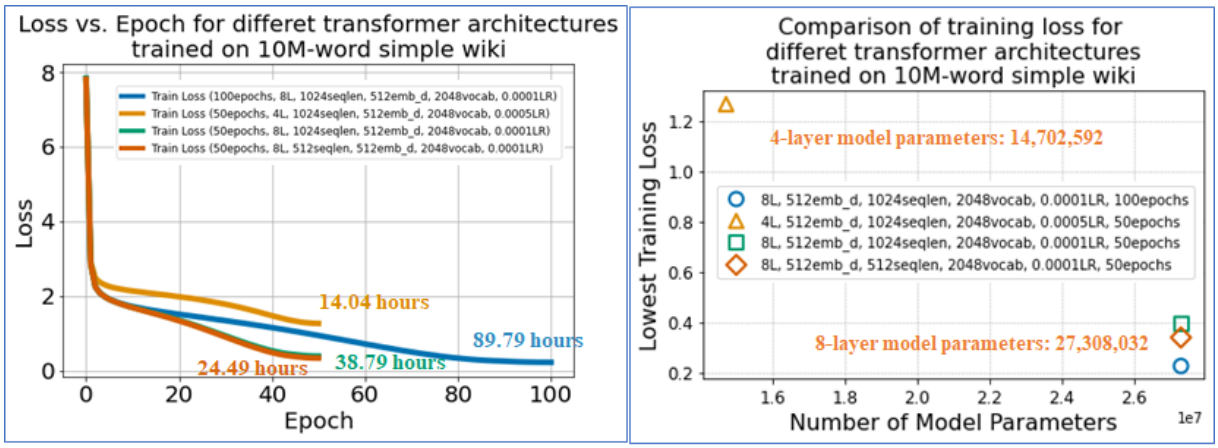


Figure 14: (a) Top: Training loss of models trained on the 10M-word subset with embedding dimensions of 256 or 512, layer counts of 1, 2, 4, or 8, and identical hyper-parameters: vocabulary size 2,048, 8 heads, and an initial learning rate of 10^{-4} with a cosine scheduler. Bottom: Best training loss from the top plot versus number of model parameters, showing that increasing the number of layers and embedding dimension reduces training loss at the cost of more parameters. (b) Top: Training loss of models trained on the 10M-word subset with initial learning rates of 1×10^{-3}, 5×10^{-4}, 1×10^{-4}, or 5×10^{-5}, layer counts of 4 or 8, and identical hyper-parameters: vocabulary size 2,048, embedding dimension 512, and 8 heads. Bottom: Best training loss from the top plot versus number of model parameters, showing that the 8-layer model with an initial learning rate of 1×10^{-4} achieves the lowest loss (1.2919).



Lowest training loss: 0.2259, from 8L, 512emb_d, 10^{-4} learning rate, at epoch 100.

Figure 15: (Left) Training loss of models trained on the 10M-word subset with embedding dimensions of 256 or 512, 4 or 8 layers, and varying numbers of epochs, using identical hyper-parameters: vocabulary size 2,048, 8 heads, and an initial learning rate of 10^{-4} with a cosine scheduler. (Right) Best training loss from the left plot versus number of model parameters. The 8-layer model trained for 100 epochs achieves the lowest loss of 0.2259.

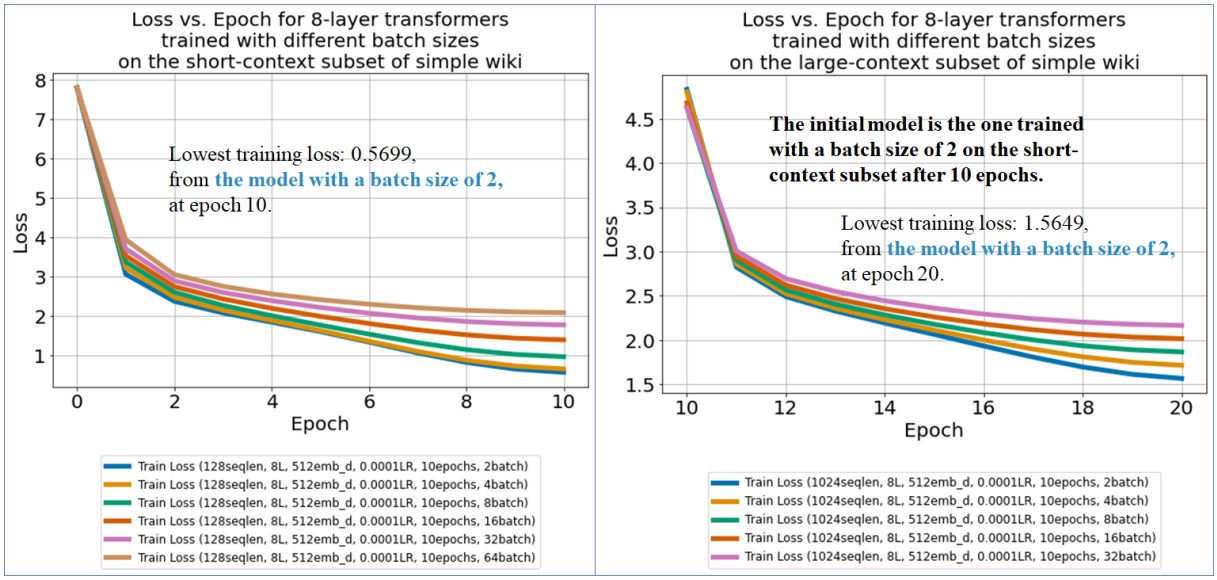


Figure 16: (Left) Training loss of the same model trained on the short-context subset with batch sizes ranging from 2 to 64. (Right) Training loss of the same models evaluated on the long-context subset after training on the short-context subset with different batch sizes. The model trained with a batch size of 2 achieves the lowest loss of 0.5699 on the short-context subset and 1.5649 on the long-context subset. All models share the same architecture: embedding dimension 512, 8 layers, vocabulary size 2,048, and 8 heads, trained with an initial learning rate of 10^{-4} using a cosine scheduler.

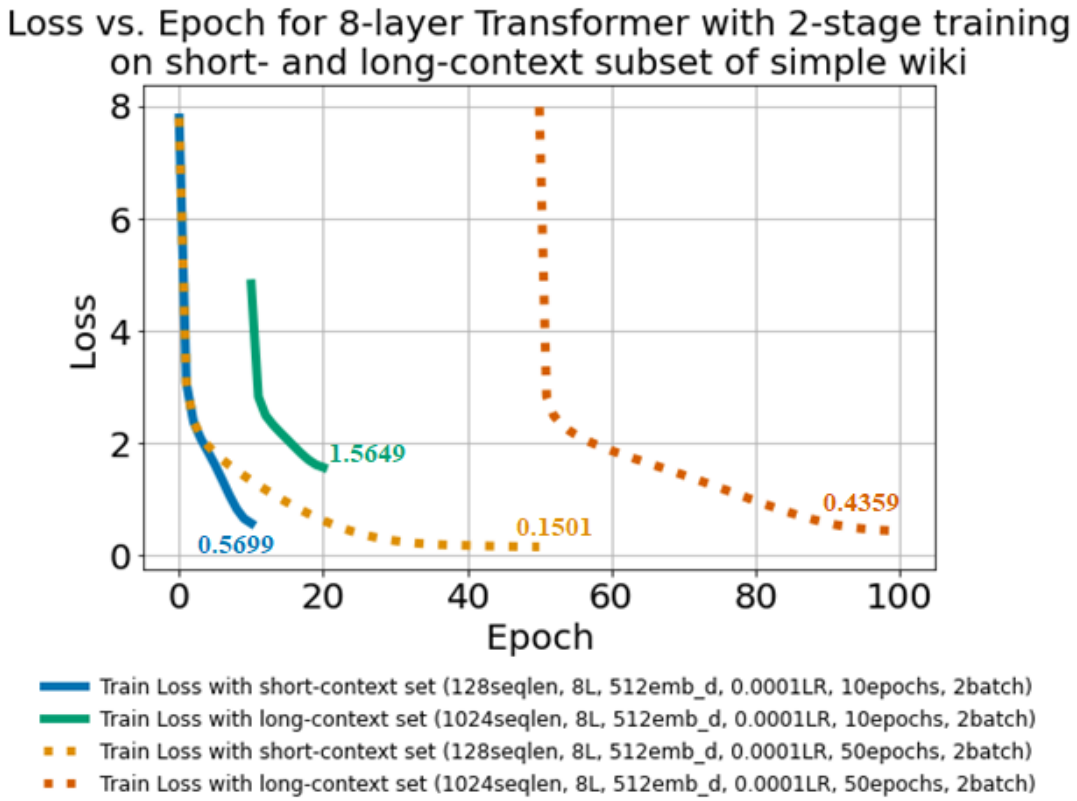


Figure 17: Training loss of 8-layer transformer models from 2-stage training with different numbers of epochs. Solid lines indicate training for 10 epochs on the short- and long-context subsets, and dotted lines indicate training for 50 epochs. The 50-epoch model achieves lower losses (0.1501 on the short-context subset and 0.4359 on the long-context subset) compared to the 10-epoch model, indicating that increasing the number of training epochs improves performance in 2-stage training.

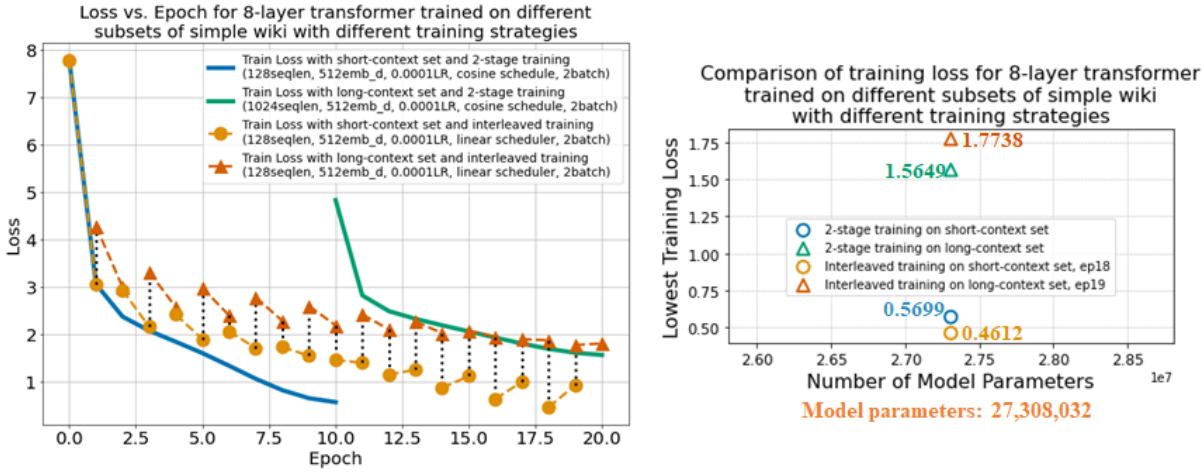


Figure 18: (Left) Training loss of 8-layer transformers with 2-stage (solid) and interleaved (dashed) training. (Right) Best training loss versus parameter count. Interleaved training achieves a lower training loss on the short-context subset (0.4612 compared to 0.5699 for 2-stage training) but a higher training loss on the long-context subset (1.7738 compared to 1.5649 for 2-stage training).

What is the Best Sequence Length for BabyLM?

Suchir Salhan[*] 🍋🍊 **Richard Diehl Martinez**[*] 🍋
Zébulon Goriely 🍋 **Paula Buttery** 🍋🍊
🍋 Department of Computer Science & Technology, University of Cambridge, U.K.
🍊 ALTA Institute, University of Cambridge, U.K.
{sas245, rd654, zg258, pjb48}@cam.ac.uk

Abstract

Transformer language models typically operate with a fixed-length context window, which has grown in step with large-scale pretraining datasets. In the BabyLM Challenge, however, many past submissions have defaulted to using much shorter sequence lengths.

We examine the impact of sequence length on BabyLM pretraining, to answer the simple question: what sequence length should we be using when training Baby LMs? Using 100M-word training data and fixed compute budgets, we compare 125M-parameter Mamba and OPT models, finding that although longer is often better, the optimal length depends on both task and architecture. Shorter sequences are sufficient for grammatical generalization tasks whereas longer contexts benefit morphological analogical reasoning tasks.

How Long can You Go? on HuggingFace (models, tokenizers, and checkpoints)

Training Code Open-Sourced on GitHub

1 Introduction

Transformer language models typically operate with a fixed context window, which has expanded in step with the growth of pre-training datasets — from millions (Kiros et al., 2015) to trillions (Soldaini et al., 2024) of tokens. Larger windows have improved performance on long-sequence reasoning tasks such as HellaSwag (Zellers et al., 2019) and MMLU (Hendrycks et al., 2020).

The BabyLM Challenge (Charpentier et al., 2025) encourages researchers to revisit foundational assumptions in language-model pretraining. In this setting, models train on a 100M-token corpus, which may be repeated up to ten times for a total of 1B tokens. Under these constraints, the belief that "longer context is always better" is less certain. Prior submissions to the challenge typically make use of shorter sequence lengths (Warstadt et al., 2023), often in an attempt to avoid training instability given the restricted data and as a cognitively-inspired attempt to mimic human working memory limitations (Cheng et al., 2023).

Our starting question is simple: what happens if we train a BabyLM using the same methods typically applied at large scale? Many submissions implicitly assume that small batch sizes and short sequences are both cognitively plausible and optimal under limited data. But is this actually true?

The Case for Long Sequences The main benefit of training language models with longer sequence lengths is **training efficiency**. Longer sequence lengths allow the model to observe more tokens per step, provide more learning signal per update, and reduce the noise in gradient estimates.

The Case for Small Sequences However, in the data constrained setting of the BabyLM challenge, using larger sequences means models are updated less often; smaller sequences, despite yielding noisier gradient approximations, enable models to be updated more overall.

These trade-offs motivate our first research question: **what is the optimal sequence length for each BabyLM evaluation task?** We explore optimality both in terms of the sequence length that produces the highest score at the end of training, as well as a more nuanced analysis that considers training time.

Next, we explore a second related question: **does this optimal length depend on the model *architecture*?** State Space Models (SSMs) are particularly interesting here: by removing the n^2 state-storage requirement of self-attention, they may handle long sequences more efficiently than Transformers.

To investigate, we train two BabyLM families—one using the Open Pre-Trained Trans-

[*]Equal contribution

Proceedings of the First BabyLM Workshop, pages 130–146
November 8, 2025

former (OPT) (Zhang et al., 2022), the other using Mamba (Gu and Dao, 2024)—on the 100M STRICT BabyLM dataset, varying only input sequence length. We span short contexts (64 tokens) common in cognitively-inspired setups to very long contexts (8192 tokens) typical in modern LLMs.

Results show that the ideal sequence length for training language models depends heavily on both the specific task and the model architecture. For some tasks, such as syntactic evaluation benchmarks, shorter sequences provide better performance and faster training times. In contrast, tasks that require understanding longer context, like entity tracking or reading comprehension, benefit from much longer sequences, sometimes up to 8192 tokens. When comparing model architectures, we find that the OPT Transformer generally performs best with a wider range of sequence lengths, including very long contexts, while the Mamba state-space model tends to achieve near-optimal results using shorter or moderate-length sequences. This suggests that different sequence-length strategies may be needed depending on the model's design and the nature of the task. We provide a set of sequence length recommendations for BabyLM practioners aiming to balance training efficiency and model performance. Selecting a training sequence length tailored to the specific task and model architecture can significantly reduce computational costs and training time without sacrificing accuracy, with the added benefit of making pretraining BabyLMs more accessible and environmentally friendly.

2 Background

2.1 Sequence Length and Modern Language Models

Multiple studies suggest that shorter sequence lengths can benefit smaller language models, particularly under data constraints. In the BabyLM setting, Cheng et al. (2023) report that using individual sentences and avoiding sequence packing yields better results, with sequences as short as 32 tokens outperforming 512-token contexts. Warstadt et al. (2023) similarly note that many top submissions to BabyLM used short contexts, aligning with developmental-learning constraints and maximizing limited data efficiency.

Outside BabyLM, compute-efficient training approaches also favor short sequences. Both Izsak and Berend (2021) and the original BERT work (Devlin et al., 2019) train primarily with 128-token sequences before a final phase at 512 tokens, while Geiping et al. (2023) find 128 tokens sufficient for strong downstream performance even with larger datasets. The LTG-BERT model from the first BabyLM Challenge adopts the same 128-to-512 token schedule (Samuel et al., 2023).

2.2 Sequence Length Across Architectures: Transformers and State-Space Models

Sequence length L plays different roles across architectures. In Transformers, L defines a fixed input window for both training and inference, directly determining attention cost. Inputs longer than the maximum L must be truncated or handled with long-context extensions such as structured attention (Hao et al., 2022) or compression (Li et al., 2023). Length extrapolation methods adjust positional embeddings to process sequences beyond the trained L (Press et al., 2021; Chen et al., 2021; Su et al., 2024), while interpolation integrates new information into existing positions (Chen et al., 2023).

By contrast, recurrent models and State Space Models (SSMs) such as Mamba do not impose a hard cap on L. Mamba retains memory via parameterized state-space dynamics, capturing long-range dependencies with linear scaling (Gu and Dao, 2024). Trained with sequences up to $L = 2048$, it can carry compressed history across chunks, making long contexts less costly in memory and computation. These differences suggest that Mamba may have a higher training-optimal L than a vanilla Transformer like OPT, owing to its more efficient handling of long-range information.

2.3 Sequence Length, Working Memory, and Psychometric Plausibility

The use of shorter sequence lengths aligns with findings in cognitive modeling. A central idea in Cognitive Science is that working-memory limitations can, paradoxically, aid language learning by imposing a recency bias and promoting abstraction through chunking (Newport, 1988; Christiansen and Chater, 2016; Wilcox et al., 2025).

Elman (1990) showed that recurrent neural networks trained on simple, short sequences in early learning stages were better at acquiring syntactic generalizations. This "starting small" strategy reflects two hypotheses: (i) learners may benefit from gradually increasing input complexity rather than starting with long or complex sequences (Ben-

gio et al., 2009), a principle used in Curriculum Learning approaches to the BabyLM Challenge (Diehl Martinez et al., 2023; Salhan et al., 2024); and (ii) memory limitations act as a resource constraint, forcing language input to be "chunked" into storable, manipulable units. This second view has motivated BabyLM approaches that incorporate cognitively inspired working-memory constraints (Armeni et al., 2022; Mita et al., 2025; De Varda and Marelli, 2024; Thamma and Heilbron, 2025; Clark et al., 2025). For example, Thoma et al. (2023) adopt a maximum sequence length of 512 for their CogMemLM architecture.

3 Methodology

We train OPT and Mamba models on the STRICT 100M subset of the BABYLM corpus (Charpentier et al., 2025) using sequence lengths ranging from 64 to 8192 tokens. Our goal is to identify a sequence length L^* that balances task performance with computational efficiency.

3.1 Default Model Hyperparameters

Parameter	Mamba	OPT
`vocab_size`	50257	50272
`hidden_size`	768	768
`num_hidden_layers`	32	12
`state_size`	16	–
`expand / ffn_dim`	2	3072
`num_attention_heads`	–	12
`hidden_act`	silu	relu

Table 1: Key default hyperparameters for `MambaConfig` and `OPTConfig` as implemented in Hugging Face Transformers.

We include a full table of training hyperparameters in *Table* 3.

3.2 Model Families

We train two model families: one based on the OPT architecture and the other on Mamba. A custom tokenizer is trained on the full BabyLM training set, starting from the Byte-Pair Encoding (BPE)-based GPT-2 tokenizer provided by Hugging Face (Sennrich et al., 2016), then retrained on the BabyLM dataset. For each model family, we train models with and without warmup. In our warmup models, we scale the learning rate linearly with sequence length, using 64 tokens as a reference, to maintain approximately constant per-token updates across sequences from 128 to 4096 tokens, and increase it gradually from zero during a warmup period to stabilize early training. We follow the checkpointing logic required for submission models in the 2025 Shared Task (Charpentier et al., 2025), saving checkpoints at increasingly intervals.

3.3 Dataset Preparation

The BABYLM training corpus is shuffled at the document level, tokenized, and split into fixed-length chunks matching the target sequence lengths: 64, 128, 256, 512, 1024, 2048, 4096, and 8192 tokens. This produces eight distinct datasets, one for each sequence length. Key hyperparameters for the two model configurations are listed in Table 1 and we open-source our trained models and the eight prepared datasets.[1]

3.4 Training-Optimal Sequence Length

Our setup allows us to examine the trade-off between sequence length, task performance, and computational cost in a controlled manner. Let $M(L)$ denote a BabyLM model trained with sequence length L, and E a BabyLM evaluation task. If two models $M(L_1)$ and $M(L_2)$ achieve comparable accuracy on E, but $T(M(L_1)) \ll T(M(L_2))$ in training time, we consider $M(L_1)$ the more *training-optimal* choice for E.

We define the **training-optimal sequence length** L^* for task E as the shortest L that yields competitive accuracy relative to other lengths while offering a measurable training-time benefit. Training time is expressed as a proportion of the longest run within the same model family to facilitate comparison under setup variance and without exhaustive hyperparameter sweeps.

3.5 Evaluation

We report L^* for each model family (OPT and Mamba) and each evaluation task in the BabyLM Evaluation Pipeline. This addresses two research questions:

1. **Task-level trends:** Do values of L^* show consistent patterns across BabyLM evaluation tasks E?

2. **Architecture-level trends:** Do differences in L^* between Mamba and OPT reflect their distinct sequence-handling mechanisms, as discussed in Section 2.2?

[1] *url anonymized for review.*

While a single L^* that improves performance across all tasks is unlikely, some practitioners may wish to optimize for overall leader board performance (e.g., maximizing the "text-average" score across zero-shot tasks), whereas others may target specific benchmarks such as BLiMP (Warstadt et al., 2020) or *psychometric fit*. The latter, introduced in the 2025 Shared Task (Charpentier et al., 2025), comprises two tasks:

- **Wug Adjectival Nominalisation** (Hofmann et al., 2025) — tests morphological analogical generalisation, e.g., AVAILABLE → AVAILABILITY.
- **Readability Prediction** (de Varda et al., 2024) — evaluates model alignment with human processing by correlating cloze probabilities with human predictability ratings from self-paced reading and eye-tracking data.

4 Results

4.1 Optimal Sequence Length, $L*$, for BabyLM Evaluation Task

In *Figure* 1, we plot the training time for OPT model with different sequence lengths. This shows accuracy of eight OPT 125M parameter models trained on the 100M STRICT corpus across training, plotted against the training time for each model. The figure only shows results for the OPT family with warmup (see *Table* 6 for full results). Using the training time data, we can identify the *training-optimal sequence length from the OPT model family* L^*_{OPT} for each BabyLM evaluation task by selecting the shortest sequence length that still achieves near-peak performance.

The effect of sequence length is task-dependent across BabyLM Evaluation Tasks. We find that the effect of sequence length is inconsistent across tasks in the 2025 BabyLM Evaluation Pipeline (Charpentier et al., 2025). There is a non-monotonic benefit of sequence length.

General Trends. Shorter sequence lengths perform better on **BLiMP** and **BLiMP Supplement**. The best performance on BLiMP is obtained by our `opt-256` model, while `opt-64`, `opt-128` and `opt-256` obtain similar performance on BLiMP Supplement, with performance generally declining as sequence length increases beyond 1024 tokens.

Our shortest sequence length model `opt-64` obtains the highest accuracy on the **EWoK** benchmark, however, it remains largely stable across sequence lengths, suggesting that EWoK tasks are less sensitive to the sequence length.

Conversely, longer sequence lengths perform better on Entity Tracking, Wug and Reading Evaluation Tasks. We can an opposite pattern for BLiMP and BLiMP Supplement. For OPT, **Entity Tracking** performance shows modest sensitivity to sequence length, with no consistent upward trend as sequence length increases. While mid-range sequences (256–1024 tokens) achieve comparable scores, extreme lengths (4096–8192 tokens) exhibit more variable results, indicating that longer contexts do not reliably improve entity-tracking capabilities. However, shorter sequence length models generally perform poorly on the Entity Tracking task, with `opt-256` achieving an accuracy of 32.42%.

For OPT, performance on the **Wug** evaluation task strongly benefits from longer sequence lengths, particularly at 4096–8192 tokens with warmup, where accuracy reaches up to 90%. This suggests that **longer sequence lengths might support learning productive morphological patterns and generalizing to novel forms**.

Overall, these results indicate that OPT's optimal sequence length is highly task-dependent: shorter sequences support better BLiMP performance, whereas longer sequences support lexical productivity tasks, like Wug, and Entity Tracking.

4.2 Model Architecture: Mamba and OPT

We similarly report L^*_{Mamba} for each BabyLM evaluation task. Scaled training time-accuracy curves for our Mamba Family are shown in *Figure* 2. Table 2 shows the training-optimal sequence lengths (L) and the lengths yielding the best evaluation performance (L_{best}) for OPT and Mamba across BabyLM tasks, alongside training cost relative to the longest-context setting.

Mamba achieves slightly lower performance than OPT across most benchmarks, often matching or slightly exceeding OPT on mid-range context tasks, while OPT tends to dominate in long-context tasks. For instance, on BLiMP and BLiMP Supplement, Mamba reaches comparable scores to OPT despite shorter sequence lengths, but in general, performance is lower than OPT. On Entity Tracking, a long-range dependency task, Mamba performs best at sequence lengths of 128–1024 tokens, whereas OPT benefits from much longer contexts (up to 8192 tokens). However, again, performance is generally lower than OPT. On Wug and EWoK,

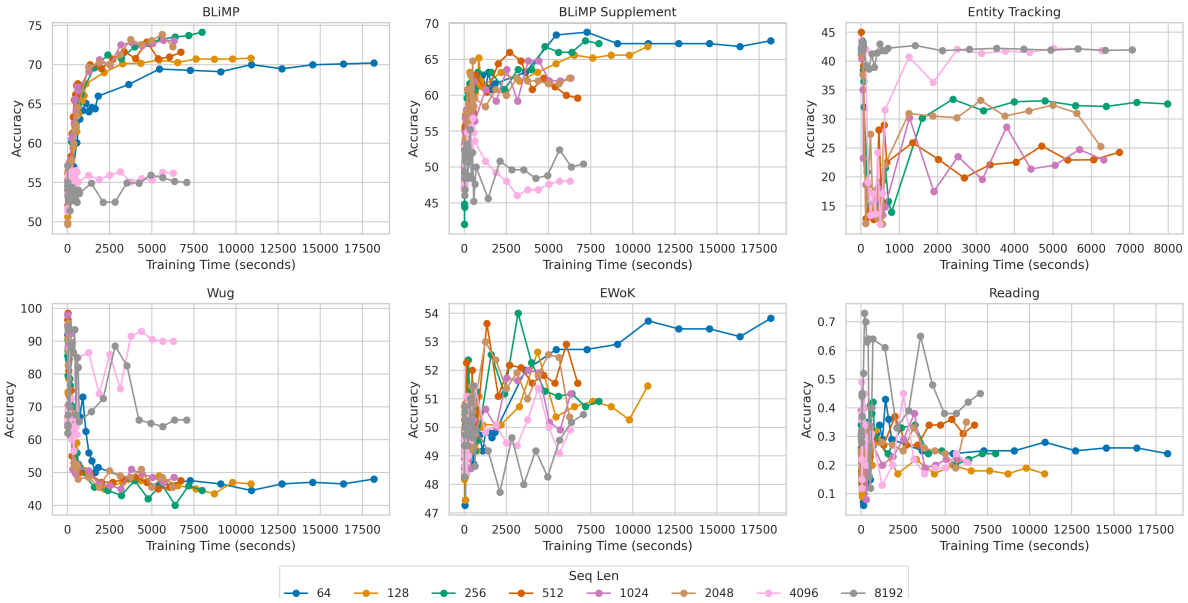


Figure 1: OPT Model Families: Effect of Sequence Length Accuracy vs Training Time per Metric. Evaluation of OPT 125M Family trained on 100M STRICT BabyLM Corpus with Warmup with a range of sequence lengths $\{64, 128, 256, 512, 1024, 2048, 8192\}$ on the Zero-Shot Evaluation Tasks of the 2025 BabyLM Evaluation Pipeline (Charpentier et al., 2025)

Task	**OPT**				**Mamba**			
	L^*	% (Longest)	L_{best}	% (Longest)	L^*	% (Longest)	L_{best}	% (Longest)
BLiMP	1024	34.8	64	100.0	512	37.3	2048	33.3
BLiMP Suppl.	256	43.9	64	100.0	64	100.0	64	100.0
Entity Tracking	4096	34.5	8192	38.8	1024	35.2	128	58.4
Wug	4096	34.5	4096	34.5	128	58.4	128	58.4
EWoK	4096	34.5	2048	34.3	1024	35.2	512	37.3
Reading	8192	38.8	8192	38.8	512	37.3	64	100

Table 2: Training-optimal sequence lengths L^* and best-performing lengths L_{best} for OPT and Mamba models on BabyLM evaluation tasks, with training time as a percentage of the longest training time for that model.

Mamba generally performs comparably to OPT at moderate sequence lengths (128–512 tokens). On Wug, Mamba outperforms OPT on nearly all sequence lengths, except the longest sequence lengths (4096). Mamba's EWoK performance is comparable to OPT but consistently obtains a marginally lower accuracy. We include a full table of results (*Table* 6) that provides a side-by-side comparison of Mamba and OPT.

The Reading results exhibit a striking pattern: Mamba achieves its peak score using the shortest context (64 tokens), whereas OPT continues to improve up to 8192 tokens. These results highlight task-specific differences in optimal context requirements between the two model families. Examining sequence length optimality, we observe that Mamba consistently prefers mid-range sequences (L between 64 and 1024 tokens) for training efficiency and evaluation performance, whereas OPT exhibits a wider spread (L between 256 and 8192 tokens).

Comparing Learning Dynamics, Mamba often attains near-peak evaluation performance with substantially shorter sequences than OPT, implying faster training times and reduced computational cost without substantial loss in accuracy. This behavior suggests that the Mamba architecture effectively leverages its hybrid attention mechanisms to capture both local and moderately long-range dependencies, reducing the necessity for extremely long contexts that OPT requires for certain tasks.

Table 2 offers actionable guidance for selecting efficient sequence lengths. For practitioners using **OPT**, we recommend $L^* = 256$ or 512 for syntax-sensitive tasks like BLiMP and BLiMP Supplement, achieving 35–44% of the full train-

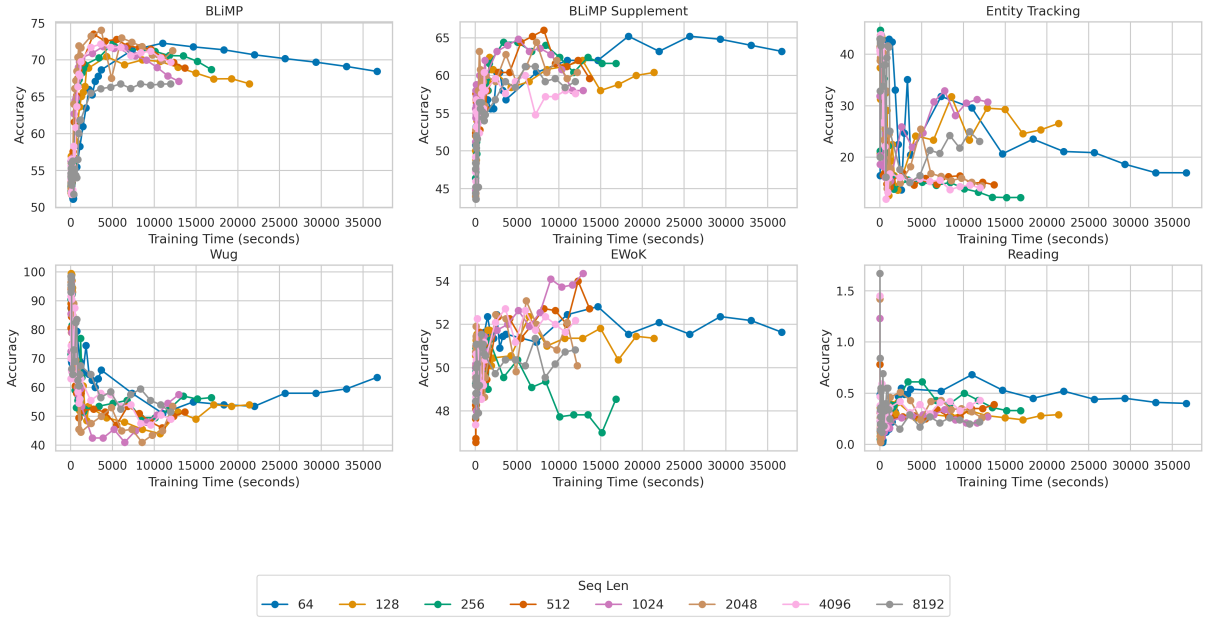


Figure 2: Mamba Model Families: Effect of Sequence Length Accuracy vs Training Time per Metric

ing cost while retaining high accuracy. For tasks requiring long-range dependencies, such as Wug, Entity Tracking, and Reading, longer contexts ($L^* = 4096$ or 8192) yield meaningful gains but at higher computational cost. Practitioners can adopt $L = 2048$ as a reasonable default for OPT to balance efficiency and generality across BabyLM tasks.

For **Mamba**, L^* values tend to cluster at shorter lengths. We recommend $L = 64$ or 128 for BLiMP Supplement, Wug, and Reading, where training time can be reduced by up to 60–65% without significant accuracy loss. Mamba's performance on EWoK and Entity Tracking is best at mid-range lengths ($L = 512$–1024), suggesting practitioners should avoid unnecessarily long contexts for most tasks. Overall, $L = 512$ offers a safe and efficient baseline across both architectures when training budget or time is limited. These recommendations allow users to reduce compute overhead while maintaining competitive task-level performance.

4.3 Psychometric Plausibility and Sequence Lengths

Figure 3 reports the evaluation of the OPT family on the readability prediction task (De Varda and Marelli, 2024).

We evaluate model performance on two psycholinguistic benchmarks—eye-tracking and self-paced reading—across varying input sequence lengths. As shown in Figure 3 (top), Mamba models exhibit relatively stable eye-tracking scores as context length increases, consistently outperforming their OPT counterparts at longer contexts (e.g., Mamba-4096 vs. OPT-4096). Notably, OPT-8192 achieves the highest accuracy ($\sim$0.45), indicating improved alignment with human eye-tracking behavior for extended inputs. In contrast, OPT models show more variable performance, with a decline in accuracy at mid-to-long sequence lengths, followed by a modest recovery at 8192 tokens.

For the self-paced reading benchmark (Figure 3, bottom), accuracy is generally lower across both model families, reflecting the greater challenge of modeling human reading times. Only the OPT-8192 configuration achieves a notable gain ($\sim$0.35), suggesting that long-context processing is critical for capturing self-paced reading patterns. While Mamba models outperform OPT at intermediate lengths (e.g., Mamba-2048 vs. OPT-2048), they fall short at the longest context window, indicating potential limitations in modeling long-range syntactic and semantic dependencies effectively.

Overall, Mamba outperforms OPT on eye-tracking prediction at long contexts, suggesting some alignment with incremental human sentence processing. However, OPT recovers and exceeds Mamba on self-paced reading at very long contexts.

5 Discussion

Our results suggest that sequence length plays a central, task-sensitive role in small-scale language

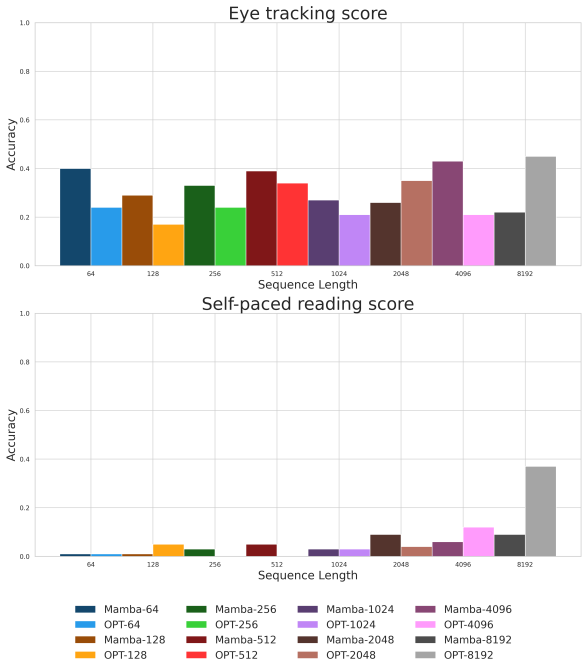


Figure 3: Distribution of Reading Sequence Length Model Accuracies for OPT Architecture

modeling, particularly within the BabyLM benchmark suite. Rather than observing a monotonic relationship between longer sequences and better performance, we find that each task exhibits a distinct profile of sequence length sensitivity. This challenges the default practice of adopting a single sequence length for all training and evaluation scenarios and suggests that per-task tuning of input length may yield significant efficiency gains without sacrificing accuracy.

5.1 Effect of Model Architecture

When comparing architectures, we find that **OPT and Mamba differ substantially in their sequence length dynamics**. The OPT family benefits from long contexts on tasks like Reading and Wug, with optimal sequence lengths (L^*) extending up to 8192 tokens. In contrast, tasks such as BLiMP and EWoK reach peak or near-peak performance at much shorter lengths (64–256 tokens).

This heterogeneity is likely task-related and reflects the diversity of BabyLM tasks. As the evaluation pipeline incorporates more tasks, there are differences in the types of linguistic structures that they emphasise—e.g., syntactic locality in BLiMP versus document-level coherence in Reading—making sequence length a proxy for task-specific inductive biases. This makes it challenging to develop one model that performs uniformly well across all tasks. Additionally, *Figure* 1 reveals pronounced fluctuations in training performance across sequence lengths, particularly for Wug and other productivity-oriented tasks. Many models show declining accuracy after initial progress, indicating that longer training does not always improve evaluation outcomes. For these tasks, shorter or mid-range sequences lengths achieve near-peak accuracy faster, reducing both computation and potential overfitting. From a practical perspective, compute-efficient training to improve performance on these tasks may involve early stopping after a moderate number of updates—around 512 steps in our experiments.

Compared to our expectations of the differences between Transformers and SSM-based architectures like Mamba, the observed OPT results only partially align. Unlike Transformers, which pay a quadratic compute and memory cost for longer contexts, and unlike RNNs, which must propagate hidden states step-by-step, Mamba's recurrence-style state updates allow it to scale more gracefully with window size. While we predicted an optimal range of 256–1024 tokens for most tasks, some OPT tasks indeed peaked in this mid-range, but others (notably Reading and Entity Tracking) favored much longer sequences than expected, suggesting certain BabyLM subtasks draw more heavily on full-document context. For Mamba, the findings diverge more strongly from our forecast. We anticipated a right-shifted optimum (1024–4096) and broad benefits from longer windows, yet L^* clustered at shorter lengths (64–1024) and L_{best} often appeared in the lower mid-range. Despite Mamba's architectural promise—continuous-time dynamics and implicit memory—we observe that **Mamba consistently prefers shorter or mid-range sequence lengths**, with L^* clustering between 64 and 1024 tokens. While this allows Mamba to train more efficiently than OPT on average, it often lags in final performance, particularly on tasks requiring sustained access to long-range dependencies (e.g., Entity Tracking, Reading). These results complicate expectations from prior work (Gu and Dao, 2024) suggesting Mamba-like models can exploit long contexts more effectively than Transformers. In our small-model, low-data regime, Mamba's theoretical capacity may be bottlenecked by optimization constraints or underutilized due to limited token diversity.

While Mamba's theoretical strengths in long-context modeling are appealing, fully realizing

these advantages may require larger models, more diverse data, or improved optimization strategies. Future work should systematically disentangle these factors to determine whether the observed limitations are fundamental to the architecture, a consequence of optimization dynamics, or an artifact of the data scale. The consistent preference of Mamba for shorter sequences raises important questions. One possibility is that this reflects an architectural limitation: despite Mamba's theoretically continuous-time, state-space recurrence dynamics, the model may be unable to store and retrieve fine-grained information over very long sequences at small model scales. Another contributing factor could be optimization challenges: gradient diversity and update counts may be insufficient in the 100M-token regime to fully exploit long-range dependencies. Finally, data-scale constraints may limit Mamba's capacity to generalize across long contexts, since small datasets provide fewer instances of extended dependency structures for learning.

These findings suggest that Mamba's efficiency – achieving near-peak performance with shorter sequences – can reduce training time and computational cost, offering a practical advantage in low-resource or small-model settings. Nevertheless, this efficiency comes at a trade-off: for tasks where long-range dependencies are critical, OPT's Transformer-based architecture remains superior, even at the expense of substantially higher training costs. This aligns with previous observations for RNN and SSM variants in small-data regimes (Haller et al., 2024), emphasizing that architectural efficiency does not automatically translate into performance gains in low-data or small-scale contexts.

In practical terms, our results offer guidance for model selection and sequence length configuration. For OPT, shorter sequences (256–512 tokens) suffice for syntax-sensitive tasks, while longer sequences (4096–8192) are beneficial for document-level and productivity tasks. For Mamba, mid-range sequences (128–512 tokens) generally balance performance and efficiency, though extreme long contexts rarely yield additional gains. When compute budgets are limited, using Mamba with shorter sequences may provide a favorable trade-off between training time and accuracy, while OPT remains the model of choice for tasks with high long-range dependency demands.

This suggests that, at small-model scale and 100M word budgets, Mamba's state-space recurrence may not fully exploit very long contexts—possibly due to limited capacity to store fine-grained long-range information, or a stronger dependence on update count and gradient diversity than hypothesized. This mismatch invites further scrutiny into how scaling laws and data regimes modulate sequence-length utility. The BabyLM setting—100M tokens and training using an architecture with 125M parameters—imposes strong bottlenecks on both parameter and data capacity. For Transformers like OPT, longer contexts may serve to increase gradient diversity and reinforce context-sensitive representations, whereas Mamba may compress or discard such information more aggressively. The result is a modest gain in training efficiency, but with diminished generalization on long-context benchmarks. These trade-offs are particularly relevant to BabyLM's goal of modeling developmentally plausible language learning with limited resources.

5.2 Sequence Length and Psychometric Plausibility

From a cognitive perspective, our sequence length results provide direct computational support for the **"starting small" hypothesis** (Elman, 1990; Newport, 1988). In *Section* 4.3, we observed that syntactic tasks like BLiMP consistently reach peak performance at shorter sequences (64–256 tokens), a pattern that suggests limiting context during learning can facilitate chunking, abstraction, and generalisation. This mirrors the cognitive insight that constrained working memory during early language exposure can promote more robust syntactic representations. Importantly, these findings are not merely incidental: they indicate that the empirical optima for sequence length in small-scale language models align with theoretically motivated cognitive constraints, showing that "starting small" can confer measurable learning advantages even in artificial systems.

Mamba's recurrent, state-based architecture provides a compelling demonstration of this principle in practice. By maintaining local state updates and implicitly emphasizing recent context, Mamba performs stably at shorter sequences, despite having the capacity for longer-range memory. This alignment between architectural design and empirical sequence length optima suggests that Mamba operationalizes a cognitively inspired inductive bias: the model leverages local context efficiently to capture syntactic regularities, providing a computational

analogue to human working memory limitations. In contrast, OPT benefits from long sequences primarily on tasks requiring document-level integration, such as Reading or Entity Tracking, highlighting how different architectures interact with sequence length in ways that parallel the cognitive distinction between local syntactic processing and global discourse comprehension.

The psycholinguistic benchmarks further reinforce this link. Mamba's locally-informed processing produces smoother, word-by-word plausibility predictions, echoing human recency effects in reading, whereas OPT's global attention facilitates retention and manipulation of hierarchical or discourse-level structures. This complementary pattern suggests that model architecture and sequence length interact to capture different aspects of linguistic cognition: recurrence-based models like Mamba naturally encode inductive biases favoring short, syntactically rich sequences, while attention-based Transformers excel when broader context is required.

For BabyLM practitioners, we hope our results provide a practical, resource-conscious strategy for selecting sequence lengths in low-resource language modeling. By computing the training-optimal length L^* for each evaluation task E, practitioners can identify the shortest sequence that delivers near-peak performance at a fraction of the training cost. This allows for more efficient model development, particularly in constrained environments where compute or wall-clock time is limited. Rather than relying on fixed defaults (e.g., $L = 512$ or $L = 2048$), users can adopt our methodology to empirically select task-appropriate sequence lengths for their architecture of choice. As we demonstrate, L^* often varies across tasks and model types, and even small adjustments can yield substantial training-time savings without sacrificing downstream accuracy.

Taken together, our findings suggest that **no single sequence length is optimal across tasks, models, or metrics**. For BabyLM, this heterogeneity means leaderboard design and evaluation strategy should account for task-specific sequence length sensitivity. For example, syntactic tasks like BLiMP reach peak performance at short sequences (64–256 tokens), whereas discourse-heavy tasks like Reading or Entity Tracking benefit from much longer contexts (up to 8192 tokens). A practical approach would be to report, for each task, performance at the training-optimal sequence length (L^*) for each model, or include a small set of task-specific lengths that capture near-peak performance. Leaderboards could also incorporate a "context-efficiency" metric, rewarding models that achieve high accuracy with shorter sequences. This would make comparisons fairer across architectures with different context preferences (e.g., OPT vs. Mamba) and better reflect model capabilities across the diverse range of BabyLM evaluation tasks.

6 Conclusion

We present a systematic evaluation of sequence length sensitivity across BabyLM tasks, comparing the Transformer-based OPT and the state-space Mamba architectures. Our findings show that no single sequence length is universally optimal: shorter sequences often suffice for syntactic benchmarks like BLiMP, while longer contexts are necessary for tasks involving lexical productivity or discourse coherence. By identifying task-specific training-optimal lengths (L^*), we provide actionable guidance for balancing performance and efficiency in low-resource settings. Our results suggest that careful tuning of sequence length—rather than scaling alone—can yield meaningful gains in both compute and accuracy.

Limitation

One limitation of our study is that we do not vary the mini-batch size or gradient accumulation strategy in conjunction with sequence length. While we vary sequence length to study its effect on task performance, it is possible to maintain a constant number of tokens per update by adjusting the mini-batch size or gradient accumulation steps. As a result, our experiments do not fully isolate the effect of sequence length from the effective batch size or the number of tokens processed per step.

Additionally, calibrating the optimal learning rate and schedule for each sequence length is challenging. Our experiments use a linear warmup proportional to sequence length, but we did not conduct exhaustive hyperparameter sweeps. It is possible that different learning rate or batch size configurations could change the relative performance of sequence lengths or architectures, and some of our reported training-optimal sequence lengths (L^*) may shift under alternative settings.

Acknowledgements

Richard Diehl Martinez is supported by the Gates Cambridge Trust (grant OPP1144 from the Bill & Melinda Gates Foundation). Suchir Salhan is supported by Cambridge University Press & Assessment. Zébulon Goriely is supported by an EPSRC DTP Studentship. This research was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council.

References

Kristijan Armeni, Christopher Honey, and Tal Linzen. 2022. Characterizing verbatim short-term memory in neural language models. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 405–424.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, et al. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *arXiv preprint arXiv:2502.10645*.

Pu-Chin Chen, Henry Tsai, Srinadh Bhojanapalli, Hyung Won Chung, Yin-Wen Chang, and Chun-Sung Ferng. 2021. A simple and effective positional encoding for transformers. *arXiv preprint arXiv:2104.08698*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Ziling Cheng, Rahul Aralikatte, Ian Porada, Cesare Spinoso-Di Piano, and Jackie CK Cheung. 2023. McGill BabyLM shared task submission: The effects of data formatting and structural biases. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 207–220, Singapore. Association for Computational Linguistics.

Morten H Christiansen and Nick Chater. 2016. The now-or-never bottleneck: A fundamental constraint on language. *Behavioral and brain sciences*, 39:e62.

Christian Clark, Byung-Doh Oh, and William Schuler. 2025. Linear recency bias during training improves transformers' fit to reading times. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7735–7747.

Andrea De Varda and Marco Marelli. 2024. Locally biased transformers better align with human reading times. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 30–36, Bangkok, Thailand. Association for Computational Linguistics.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard Diehl Martinez, Zébulon Goriely, Hope McGovern, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 112–127, Singapore. Association for Computational Linguistics.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Jonas Geiping, Micah Goldblum, Arjun Schwarzschild, Tom Goldstein, et al. 2023. Cramming: Training a language model on a single gpu in one day. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*.

Patrick Haller, Jonas Golde, and Alan Akbik. 2024. BabyHGRN: Exploring RNNs for sample-efficient language modeling. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 82–94, Miami, FL, USA. Association for Computational Linguistics.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*.

Dan Hendrycks, Christopher Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Valentin Hofmann, Leonie Weissweiler, David R Mortensen, Hinrich Schütze, and Janet B Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Peter Izsak and Gábor Berend. 2021. How to train bert with an academic budget. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, volume 28.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. 2023. Compressing context to enhance inference efficiency of large language models. *arXiv preprint arXiv:2310.06201*.

Masato Mita, Ryo Yoshida, and Yohei Oseki. 2025. Developmentally-plausible working memory shapes a critical period for language acquisition. *arXiv preprint arXiv:2502.04795*.

Elissa L Newport. 1988. Constraints on learning and their role in language acquisition: Studies of the acquisition of american sign language. *Language sciences*, 10(1):147–172.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is more: Pretraining cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 174–188, Miami, FL, USA. Association for Computational Linguistics.

David Samuel, Anders Rekstad, and Erik Velldal. 2023. Trained on 100 million words and still in shape: Bert meets british national corpus (ltg-bert). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725. Association for Computational Linguistics.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Pete Walsh, Hannaneh Hajishirzi, Noah A. Smith, Luke Zettlemoyer, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Abishek Thamma and Micha Heilbron. 2025. Human-like fleeting memory improves language learning but impairs reading time prediction in transformer language models. *arXiv preprint arXiv:2508.05803*.

Lukas Thoma, Ivonne Weyers, Erion Çano, Stefan Schweter, Jutta L Mueller, and Benjamin Roth. 2023. CogMemLM: Human-like memory mechanisms improve performance and cognitive plausibility of LLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 180–185, Singapore. Association for Computational Linguistics.

Alex Warstadt, Sweta Agrawal Mishra, Masato Yoshida, Jon Gauthier, and et al. 2023. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Ethan Gotlieb Wilcox, Michael Y Hu, Aaron Mueller, Alex Warstadt, Leshem Choshen, Chengxu Zhuang, Adina Williams, Ryan Cotterell, and Tal Linzen. 2025. Bigger is not always better: The importance of human-scale language modeling for psycholinguistics. *Journal of Memory and Language*, 144:104650.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Shu Dewan, Marjan Ghazvininejad, Sinong Gutiérrez, Lucy Hazard, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Training Setup: Hyperparameters

Table 3: Training hyperparameters for BabyLM experiments. This table summarizes model, training, checkpointing, hardware, and dataset settings.

Category	Parameter	Value / Notes
Model	Type	• OPT: 12-layer, 768 hidden, 12 heads, FFN 3072 • Mamba: 32-layer, 768 hidden
	Vocabulary size	50,257 tokens
	Max sequence length	64–16,384 tokens, varies per experiment
	Pretrained weights	Random initialization
Training	Epochs	10
	Global batch size	64 sequences
	Per-device batch size	$\frac{\text{GLOBAL\_BATCH\_SIZE}}{(\text{num\_devices} \times \text{accumulation\_steps})}$
	Gradient accumulation steps	1 (configurable via CLI)
	Learning rate	Scales linearly with seq. length if warmup: $5 \times 10^{-5} \times \frac{\text{seq\_len}}{64}$
	Tokens per batch	GLOBAL_BATCH_SIZE $\times$ seq_len
	Tokens per update	Tokens per batch × accumulation steps
Checkpointing	Frequency	Every 1M, 10M, 100M tokens (CustomCheckpointingCallback)
	Hub push	Optional via CLI
	Resume from checkpoint	Supported
Hardware / Precision	Devices	4 (configurable via CLI)
	Mixed precision	bf16 (DeepSpeed / Trainer)
	DeepSpeed	Optional, stage 3 ZeRO with CPU offload
Dataset	Source	Hugging Face pretokenized datasets
	Examples	babylm-seqlen/train_100M_<seq_len>_single_shuffle
	Preprocessing	Labels set as input_ids for causal LM training

B Dataset Statistics

Sequence Length	Num Sequences
64	2,556,406
128	1,278,130
256	639,002
512	319,435
1024	159,656
2048	79,761
4096	39,814
8192	19,844
16384	9,863

Table 4: Number of sequences for each fixed sequence length dataset. Sequence lengths are clickable links to the corresponding Hugging Face dataset.

Table 5: Example settings for per-device batch size, learning rate, and tokens per batch at different sequence lengths.

Seq Length	**Per-Device Batch**	**Learning Rate**	**Tokens per Batch**
64	16	5e-5	4,096
128	16	1e-4	8,192
512	16	4e-4	32,768
2048	16	1.6e-3	131,072
8192	16	6.4e-3	524,288

C Final Checkpoint Results: OPT and Mamba (± Warmup)

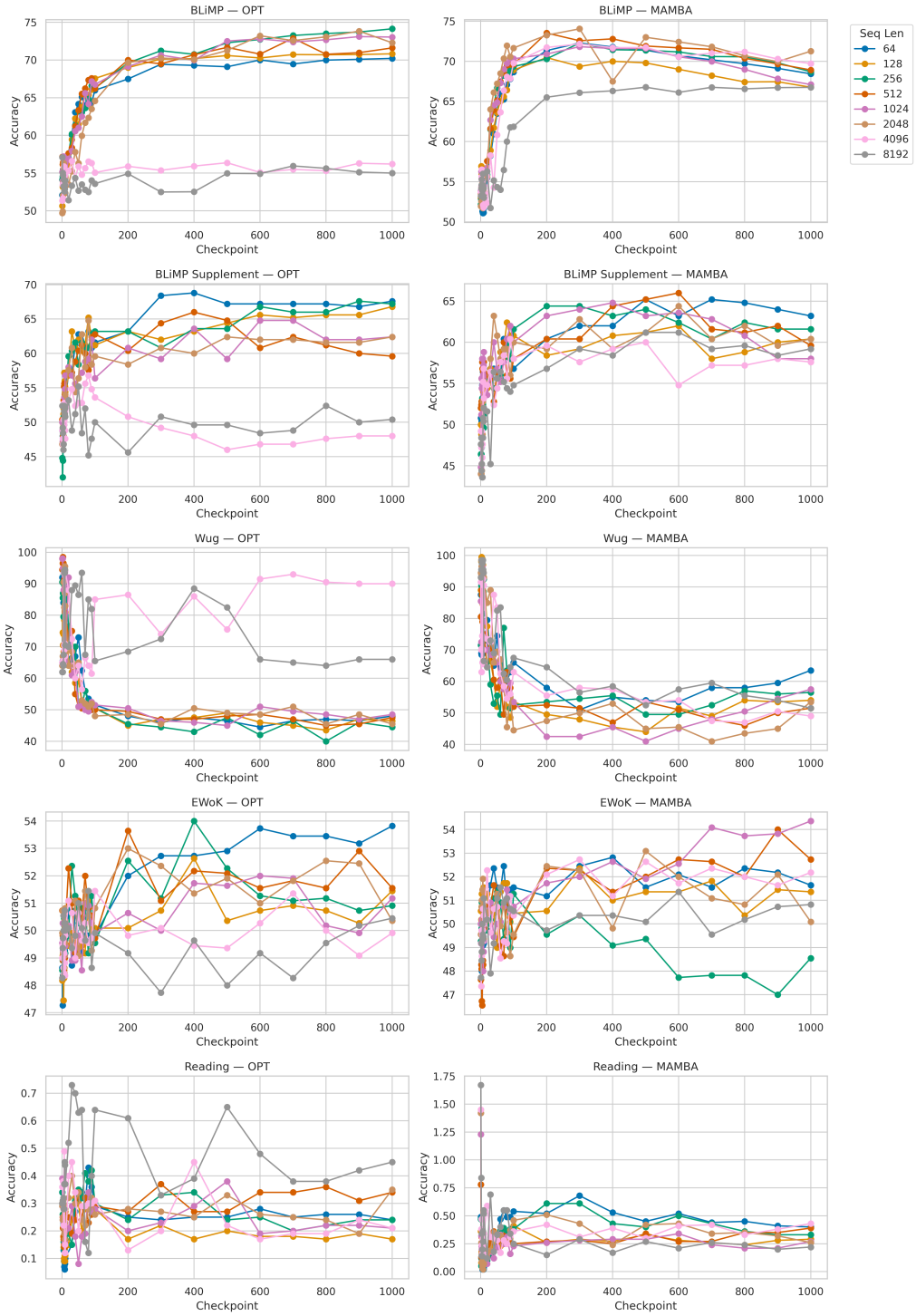
Table 6 provides a detailed breakdown of model performance on the full zero-shot evaluation tasks. In particular, we report differences between training models with and without warmup.

Model	Warmup	Seq Len	BLiMP	BLiMP Suppl.	Entity Tracking	EWoK	Wug
mamba	+	64	68.33	63.20	19.05	50.66	63.50
mamba	-	64	69.56	61.23	22.24	51.05	62.00
mamba	+	128	67.31	60.40	23.20	50.1	54.00
mamba	-	128	69.87	57.94	38.69	51.34	70.50
mamba	+	256	69.19	61.60	12.48	51.34	56.50
mamba	-	256	69.14	60.04	25.28	51.28	53.50
mamba	+	512	68.87	59.60	16.33	52.54	51.50
mamba	-	512	68.45	60.98	23.16	49.99	55.50
mamba	+	1024	67.30	58.00	31.95	52.31	57.50
mamba	-	1024	66.28	56.99	21.52	50.35	62.50
mamba	+	2048	71.62	60.40	14.07	51.82	53.50
mamba	-	2048	63.33	55.03	20.25	50.30	56.50
mamba	+	4096	69.56	57.60	13.93	51.49	49.00
mamba	-	4096	59.10	55.50	17.80	50.18	62.00
mamba	+	8192	66.91	59.20	22.70	51.05	51.50
mamba	-	8192	59.21	52.94	23.37	49.83	61.50
opt	+	64	70.21	67.60	–	51.82	48.00
opt	-	64	75.44	66.45	–	51.64	49.50
opt	+	128	70.78	66.80	–	51.92	46.50
opt	-	128	74.87	63.53	–	51.98	45.00
opt	+	256	73.88	67.20	32.42	52.18	44.50
opt	-	256	73.11	59.92	20.93	51.68	46.00
opt	+	512	71.9	59.60	26.80	51.45	47.50
opt	-	512	70.63	61.70	26.99	51.80	47.00
opt	+	1024	72.69	62.40	26.15	51.28	48.5
opt	-	1024	68.23	57.79	26.27	50.66	50.00
opt	+	2048	72.05	62.40	25.96	52.37	45.50
opt	-	2048	61.67	57.23	29.57	49.89	50.50
opt	+	4096	56.25	48.0	40.23	49.70	90.00
opt	-	4096	58.58	54.58	17.03	50.10	66.00
opt	+	8192	55.05	50.40	40.38	50.89	66.00
opt	-	8192	56.01	53.21	19.38	49.70	64.50

Table 6: Evaluation results across multiple benchmarks for Mamba and OPT models. '–' denotes missing data (NaN).

D Learning Dynamics: Task Evaluation on Checkpoints

Figure 4: Comparison of the performance of Mamba and OPT models on BabyLM Evaluation tasks throughout training. Checkpoints are saved at increasingly intervals throughout training: every 1M words until 10M words are seen, every 10M words until 100M words are seen, and every 100M words until 1B words are seen.



E Subtask Accuracy for OPT and Mamba Families

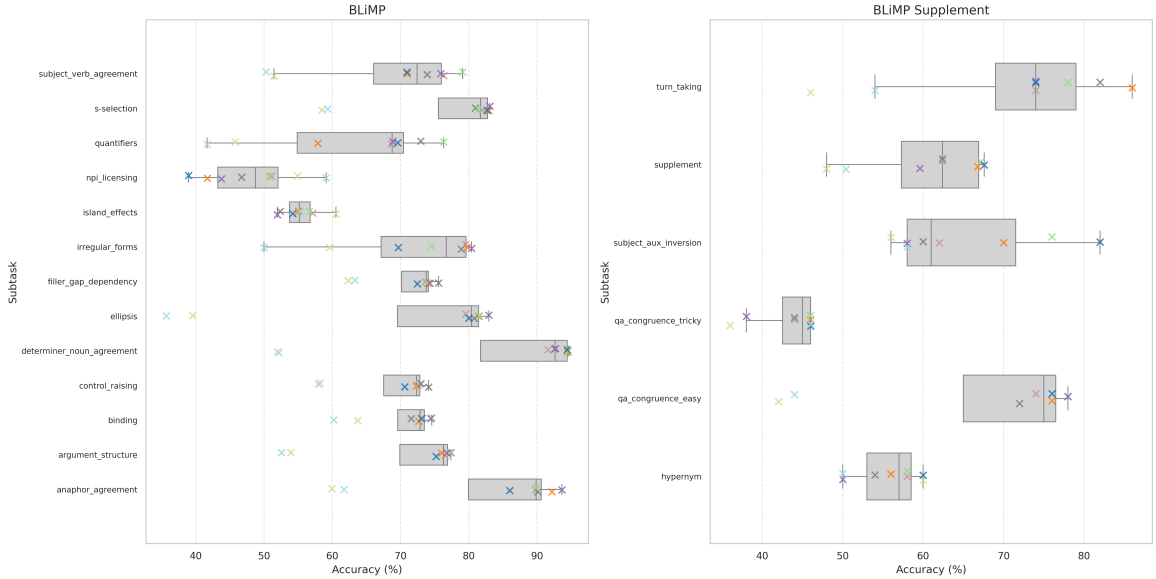


Figure 5: Distribution of OPT Sequence Length Model Accuracies on BLiMP and BLiMP Supplement

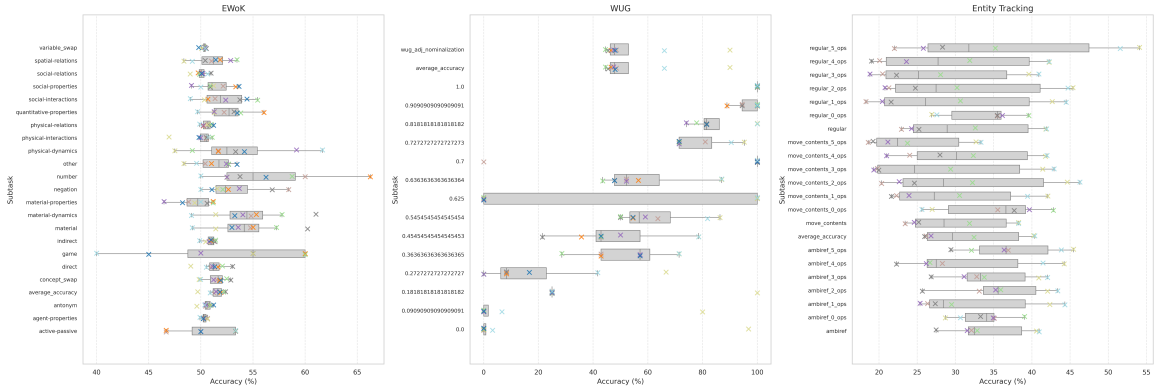


Figure 6: Distribution of EWoK, Wug and Entity Tracking Sequence Length Model Accuracies for OPT Architecture

F F1 Scores for Fine-Tuning

Figure 7: **F1 for Fine-Tuned Models**

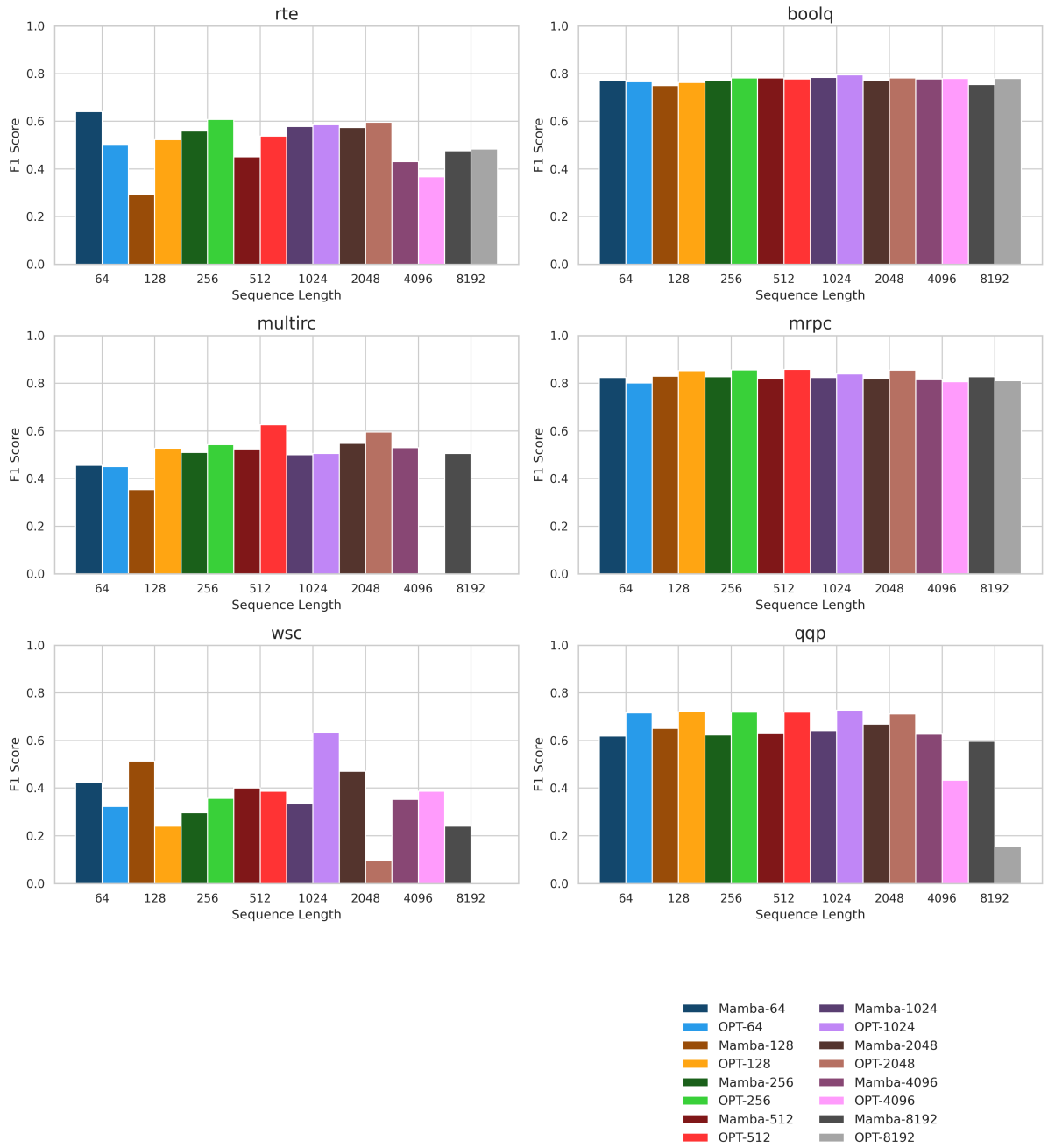


Figure 8: F1 Scores for OPT and Mamba Families on Fine-Tuned Tasks

BitMar: Low-Bit Multimodal Fusion with Episodic Memory for Edge Devices

Euhid Aman
NTUST Taiwan
M11315803@mail.ntust.edu.tw

Esteban Carlin
NTUST Taiwan
M11302809@mail.ntust.edu.tw

Hsing-Kuo Pao
NTUST Taiwan
pao@mail.ntust.edu.tw

Giovanni Beltrame
Polytechnique Montréal
giovanni.beltrame@polymtl.ca

Ghaluh Indah Permata Sari
NTUST Taiwan
d11115804@mail.ntust.edu.tw

Yie-Tarng Chen
NTUST Taiwan
ytchen@mail.ntust.edu.tw

Abstract

Cross-attention transformers and other multimodal vision-language models excel at grounding and generation; however, their extensive, full-precision backbones make it challenging to deploy them on edge devices. Memory-augmented architectures enhance the utilization of past context; however, most works rarely pair them with aggressive edge-oriented quantization. We introduce BitMar, a quantized multimodal transformer that proposes an external human-like episodic memory for effective image-text generation on hardware with limited resources. BitMar utilizes 1.58-bit encoders, one for text (BitNet-style) and one for vision (DiNOv2-based), to create compact embeddings that are combined and used to query a fixed-size key-value episodic memory. During vector retrieval, the BitNet decoder applies per-layer conditioning, which increases the contextual relevance of generated content. The decoder also employs attention sinks with a sliding-window mechanism to process long or streaming inputs under tight memory budgets. The combination of per-layer conditioning and sliding-window attention achieves a strong quality–speed trade–off, delivering competitive captioning and multimodal understanding at low latency with a small model footprint. These characteristics make BitMar well-suited for edge deployment.

Keywords: TinyVLM, Episodic memory, EdgeAI, Quantization.

1 Introduction

Visual Language Models (VLMs) have made rapid progress in recent years, excelling at tasks such as image captioning (Chen et al., 2015), visual question answering (Anderson et al., 2018; Li et al., 2022). Large-scale architectures such as BLIP-2 (Li et al., 2023), Flamingo (Alayrac et al., 2022), and Kosmos-2 (Peng et al., 2023) demonstrate that cross-attention transformers can synchronize modalities for grounded language generation. However, their full-precision, extensive backbones incur significant computational and memory expenses, which restricts their implementation on devices with resource limitations.

A growing body of work targets efficient multimodal processing, such as low-bit quantization (Dettmers et al., 2021; Frantar et al., 2022) and compact language models (Wang et al., 2023), to reduce memory/latency. Quantized ViTs (Jacob et al., 2018; Stock et al., 2019), and self-supervised vision encoders, such as DiNOv2 (Oquab et al., 2024), lower the cost of vision. Multimodal fusion ranges from early concatenation (Lu et al., 2019) to learned query transformers (Li et al., 2023) to bridge frozen vision and language models. Memory-augmented transformers (Graves et al., 2016; Borgeaud et al., 2022) retrieve past context to improve coherence. Yet no existing tiny language model effectively unifies low-bit multimodal encoding with an episodic memory system for edge deployment.

To fill this gap, we propose a compact four-stage pipeline optimized for efficient on-device execution: (1) **1.58-bit text and vision encoders** generate lightweight, quantized embeddings; (2) **a cross-modal fusion module** aligns the modalities within a shared latent space; (3) **an episodic memory** with 512 key–value slots retrieves relevant multimodal context; and (4) **a BitNet-based decoder** conditions each transformer layer on the retrieved memory for context-aware generation. Both encoders output 128-dimensional representations, and DiNOv2's original 768-D vision features are compressed to 128-D before fusion. The fused embedding queries an episodic memory of size $K = 512$, $C = 128$, whose retrieved vectors condition each decoder layer. This architecture maintains all modules in a consistent 768-dimensional space, simplifying integration and minimizing projection overhead while ensuring low-latency, memory-efficient

Proceedings of the First BabyLM Workshop, pages 147–154
November 8, 2025

operation on edge hardware.

Our main contributions are summarized as follows:

- **Low-bit multimodal encoding framework.** We propose a unified architecture that integrates a 1.58-bit quantized BitNet text encoder with a quantized ViT-based vision encoder, enabling efficient and compact multimodal feature extraction.
- **Memory-augmented decoding mechanism.** We design a lightweight episodic memory module that retrieves contextual representations and injects them into each transformer layer through per-layer conditioning, enhancing coherence and contextual relevance during generation.
- **Edge-efficient multimodal reasoning.** We demonstrate that BitMar achieves competitive performance in image captioning and multimodal understanding under extreme compression, maintaining low latency and a minimal memory footprint suitable for on-device deployment.

2 Related Work

Different VLMs and Tiny LLM architectures have emerged that enable deployment and applications of multimodal AI on resource-constrained devices. Recent developments in small VLMs, such as H2OVL-Mississippi (0.8B parameters) (Galib et al., 2024), TinyGPT-V (Yuan et al., 2024), and MiniCPM-V (Yao et al., 2024), demonstrate that compact multimodal models can achieve competitive performance while maintaining efficient deployment characteristics. Similarly, Tiny LLMs, such as MobileLLM (Liu et al., 2024) and TinyLLM (Zhang et al., 2024), have shown that sub-billion parameter models can be quantized and optimized for their deployment on edge devices. These highlight the feasibility of on-device multimodal processing, with models providing meaningful performance while addressing security, latency, and connectivity constraints.

Furthermore, memory-augmented neural networks and language models inspired by cognitive thinking, such as humans, have also garnered significant attention for their ability to store and retrieve contextual information related to specific things across certain short periods of time. Memory-augmented neural networks (MANNs) (Graves et al., 2016), use decoupled key-value structures to store and retrieve contextual information. Recent works, such as EGO (Mattar and Daw, 2024) and selective episodic memory strategies (Mattar and Daw, 2022), have extended these ideas for flexible knowledge transfer and context-based memory access. However, these models face limitations in combining memory systems with low-bit quantized multimodal encoders, often sacrificing either memory capacity or model precision.

BitMar overcomes these challenges by integrating 1.58-bit quantization across text and vision encoders, alongside a cross-modal memory retrieval system. The design enables BitMar to store and retrieve both textual and visual context, improving memory interactions and enhancing multimodal generation tasks, all while maintaining computational efficiency for edge deployment.

3 Method

We introduce **BitMar**, a deployable quantized multimodal LM for efficient image–text generation under tight resources. The four-stage pipeline is: (1) **parallel low-bit text/vision encoders**; (2) **cross-modal fusion** in a shared latent space; (3) **context augmentation** via external episodic memory; (4) **autoregressive decoding** conditioned on fused and retrieved signals. Text uses a BitNet transformer at 1.58-bit precision; vision uses DiNOv2 features plus quantization-aware compression. Fusion aligns 768-D modality latents via lightweight attention. A fixed-size episodic memory stores prior multimodal contexts and injects retrieved vectors into the decoder per layer. Unlike classic MANNs (Graves et al., 2016), BitMar integrates cross-modal retrieval under low-bit constraints. The decoder is a BitNet-based autoregressive transformer with streaming attention via attention sinks for low-latency, long-context generation.

3.1 Text Encoders

Architecture. A 4-layer quantized Transformer (d=128, h=4) supports up to 256 tokens, balancing expressiveness and efficiency.

Quantization. *Weights:* all MHSA/FFN projections use ternary $\{-1, 0, +1\}$ with learned per-layer scales (1.58-bit). *Activations:* token-wise 8-bit using per-token max-abs scaling to $[-127, 127]$, preserving local detail and stable training/inference.

Attention sinks (streaming). With S=4 sink

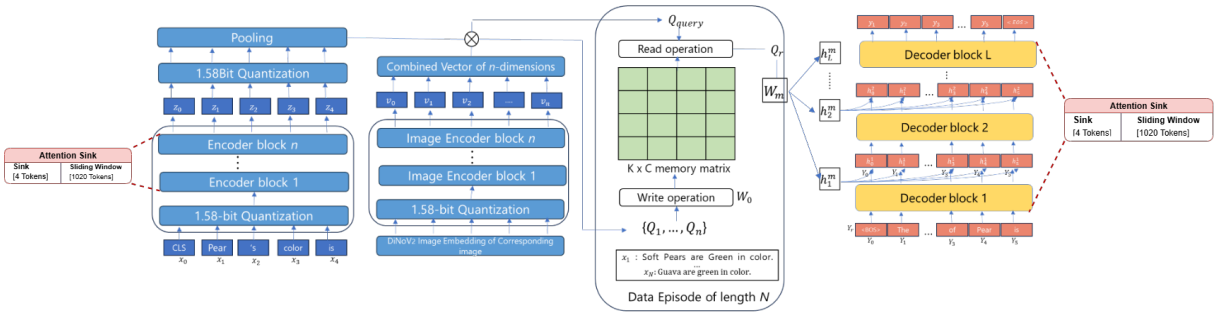


Figure 1: **BitMar Architecture.** The model processes multimodal inputs: text tokens and DiNOv2-compressed image features. Quantized encoders (1.58-bit) generate compact text and vision embeddings (z, v), which are fused via cross-modal attention into shared query representations (Q, Q_{query}). A sliding-window attention mechanism enables long-context processing. A fixed episodic memory matrix ($K \times C$) stores and retrieves multimodal context vectors through quantized read/write weights (W, W_0), supporting optional SD-card offloading for edge deployment.

tokens (never evicted) and window W=1020, the KV cache maintains persistent anchors + recent tokens. On each new token, the oldest in-window token is evicted; sink and window sets are merged; positions are clamped to $[0, S+W-1]$. This yields fixed-memory, long-context attention under low-bit compute.

3.2 Vision Encoders

We use frozen DiNOv2 (Oquab et al., 2024) to extract 768-D patch features offline, avoiding heavy vision backbones at inference. 2×2 average pooling reduces the number of patches 4× while keeping 768-D per patch. 2-layer MLP bottleneck then compresses 768→128 with ReLU and dropout between layers (parameters $\mathbf{W_1} \in \mathbb{R}^{384\times768}$, $\mathbf{W_2} \in \mathbb{R}^{128\times384}$), all subsequent fusion/memory/decoder paths operate in 128-D.

3.3 Cross-Modal Fusion

Given pooled text tokens $\mathbf{Z} \in \mathbb{R}^{n_t\times128}$ and vision tokens $\mathbf{V}_{\text{img}} \in \mathbb{R}^{n_v\times128}$, we apply standard cross-attention (Vaswani et al., 2017) (text queries, vision keys/values; cf. Transformer attention) to obtain the fused sequence $\mathbf{F} \in \mathbb{R}^{n_t\times128}$. All $Q/K/V$ and fusion projections use 1.58-bit ternary weights with learned scales; softmax and residual/LN are in FP32. We then pool $\mathbf{F}$ (mean or learned) to a single vector $\mathbf{q}_{\text{mem}} \in \mathbb{R}^{128}$ to query episodic memory.

3.4 Episodic Memory

We maintain a learnable matrix $\mathbf{M} \in \mathbb{R}^{K\times C}$ (default K=512, C=128) that stores multimodal episode vectors.

Writing. At step t, we compute a pooled query $\mathbf{q}_t \in \mathbb{R}^C$ and learned write weights $\mathbf{W}_w \in \mathbb{R}^K$. We perform soft multi-slot writes with rate α=0.2 via an outer product:

$$\mathbf{M} \leftarrow \mathbf{M} + \alpha\, \mathbf{W}_w\, \mathbf{q}_t^\top. \tag{1}$$

Reading. We use content-based addressing (Graves et al., 2016):

$$\begin{aligned} \mathbf{W}_r &= \text{softmax}(\mathbf{M}\,\mathbf{q}_t) \in \mathbb{R}^K, \\ \mathbf{M}_r &= \mathbf{W}_r^\top \mathbf{M} \in \mathbb{R}^{1\times C}. \end{aligned} \tag{2}$$

Regularization. To avoid thrashing, we penalize abrupt updates to the store with a Frobenius penalty, $\mathcal{L}_{\text{reg}} = \lambda \left\|\Delta\mathbf{M}\right\|_F^2$, $\Delta\mathbf{M} := \mathbf{M}^{(t)} - \mathbf{M}^{(t-1)}$. We additionally apply usage-based forgetting to down-weight stale slots.

3.4.1 Decoder with Attention Sinks

A 4-layer causal Transformer (d=128, h=4, max length 256) conditions on fused inputs and retrieved memory.

Long-context generation. Each layer, similarly as the text encoder, maintains KV caches of S sink tokens and a window of W recent tokens.

Memory integration. $\mathbf{M_r} \in \mathbb{R}^{1\times128}$ is projected and combined with token embeddings via either concatenation $[x_t; \mathbf{M_r}]$ (then projected) or residual addition x_t+$\mathbf{M_r}$.

Output projection. BitNet-quantized linear layer (128→50,257) maps to GPT-2 vocab logits; logits computed in FP32.

3.4.2 Training Objectives

We complement standard Language Modeling cross-entropy (Vaswani et al., 2017) and an InfoNCE cross-modal (Oord et al., 2018) term with a memory-consistency regularizer Equation 3 that

penalizes changes between successive writes to the episodic store, which discourages oscillatory updates and helps retain slot semantics. The total loss integrates these factors as Equation 4. We set $\mathcal{L}_{\text{cm}} = 1.5$ to prioritize cross-modal alignment, and $\mathcal{L}_{\text{mem}} = 0.1$ as a light stabilizer.

Memory consistency.

$$\mathcal{L}_{\text{mem}} = |\mathbf{M}_{\text{write}}^{(t)} - \mathbf{M}_{\text{write}}^{(t-1)}|_2^2 \tag{3}$$

Total objective.

$$\mathcal{L} = \mathcal{L}_{\text{lm}} + 1.5\mathcal{L}_{\text{cm}} + 0.1\mathcal{L}_{\text{mem}} \tag{4}$$

Adaptive Training Controller. When a 200-step EMA of cross-modal cosine similarity drops by > 0.12 from its recent max (with an $\geq$800-step cooldown), we randomly freeze one encoder or upweight $\mathcal{L}_{\text{cm}}$ for 1,500 steps to prevent modality collapse.

4 Experimental Setup

Our experimental framework systematically evaluates the proposed 14M-parameter BitMar model across several critical dimensions. We first benchmark its performance against established compact and low-bit baselines to assess overall viability (Table 1). We then conduct an analysis of its capabilities across a suite of language understanding and multimodal tasks to identify specific strengths and limitations (Table 2). Beyond task performance, we also investigate the internal dynamics of the model, examining how the episodic memory evolves from diffuse to structured activation patterns during training (Figure 2). Finally, we track the progression of quantization efficacy throughout the training process to validate our low-precision approach (Figure 3).

4.1 Dataset

The corpus comprises 100M tokens, split evenly between multimodal captions and text-only data.

Multimodal (50M). From CC3M (Sharma et al., 2018) and Localized Narratives (Pont-Tuset et al., 2020), aligned with precomputed DiNOv2 features (frozen backbone, reused across training).

Text-only (50M). From BabyLM (Charpentier et al., 2025), spanning six domains (BNC, CHILDES, Gutenberg, OpenSubtitles, Simple English Wikipedia, Switchboard).

Mixture. Uniform 50:50 sampling; a 1M-token hold-out tracks cross-modal alignment (cosine similarity) and perplexity.

Preprocessing. GPT-2 BPE tokenizer, max 256 tokens (truncate/pad). Visual features stored as memory-mapped ".npy" with on-the-fly compression for efficient batching.

4.2 Training Configuration

We trained on an NVIDIA A6000 GPU using FP16 and gradient checkpointing. Each step processed 64 sequences, with two-step gradient accumulation yielding an effective batch size of 128. Optimization used AdamW8bit (2×10^{-4}) with cosine restarts (T_0=1000, T_{mult}=2, η_{min}=0.1lr) for 10 epochs. We logged to Weights & Biases every 500 steps, including losses ($\mathcal{L}_{\text{lm}}, \mathcal{L}_{\text{cm}}, \mathcal{L}_{\text{mem}}$), cross-modal alignment metrics, episodic-memory utilization, attention maps, and FLOPs per step.

4.3 Hyperparameters

The model architecture employs a four-layer text encoder with 128-dimensional hidden states. The episodic memory module comprises 512 slots, each with 128 dimensions, balancing memory footprint with recall capacity. For long-context streaming, we maintain four sink tokens with a sliding window of 1020 tokens. Training utilizes weighted losses with cross-modal and memory consistency coefficients of 1.5 and 0.1, respectively. An adaptive controller triggers memory freezing when alignment metrics drop by 0.12 from their recent maximum, applying 1,500-step freezes with a minimum interval of 800 steps between interventions.

4.4 Benchmarks and Baselines

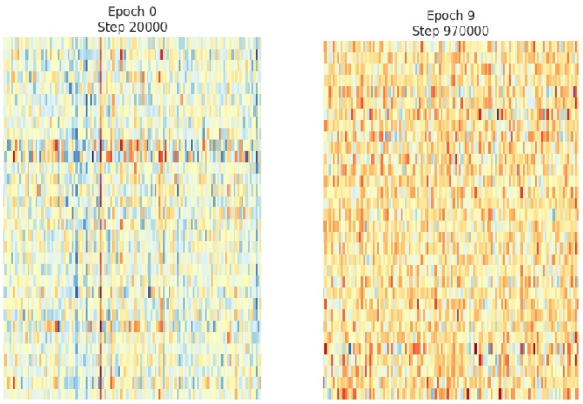
We evaluate on six language benchmarks: *ARC-Easy*, *BoolQ*, *HellaSwag*, *WinoGrande*, *CommonsenseQA*, and *MMLU*, plus multimodal tasks aligned with DiNOv2 features. Outputs are evaluated by accuracy and compared against baselines (*Bonsai 0.5B*, *OLMo-BitNet 1B*, *Falcon3-1.58bit 7B*, *LLaMA3-8B-1.58*, and *BitNet b1.58 2B*). Beyond benchmarks, we track the effectiveness of quantization and episodic activations to assess representational efficiency and memory use.

5 Results and Discussion

5.1 BitMar's performance

Figure 2 shows episodic memory slot activations over training. Early on Figure 2(a), activations are weak and scattered, with minor specialization or proper storage. By late training Figure 2(b), activations strengthen and differentiate, indicating se-

lective storage of contextual features. This progression demonstrates that extended joint optimization enables the memory to evolve into a more structured, capacity-efficient component for long-term context integration.



(a) (b)

Figure 2: **Episodic Memory Activation Patterns.** (a) Early training shows scattered and weak activations with minimal specialization. (b) Late training exhibits stronger and more differentiated activations, reflecting the emergence of structured memory representations.

We measure the quantization effectiveness E_q, inspired by (Zhu et al., 2016), as the zero-weight fraction in ternary weights across BitNet-quantized layers, where a higher value means more compression.

As training progresses (Figure 3), E_q gradually increases and stabilizes at 42.8%, demonstrating effective compression without degrading downstream performance.

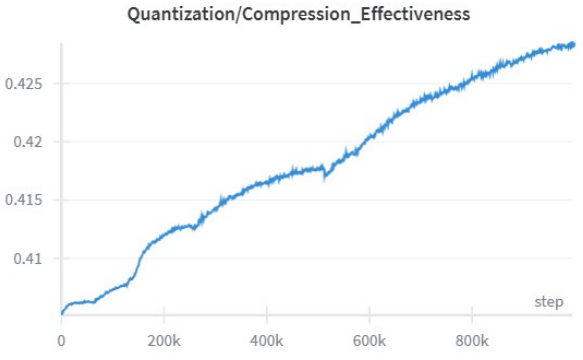


Figure 3: **Quantization effectiveness over training epochs.**

Table 1 compares BitMar-14M with low-bit baselines. Despite its small size (14M parameters), it achieves competitive performance on *BoolQ* (42.8) and *WinoGrande* (54.6), demonstrating strength in binary reasoning and coreference. On *ARC-Easy* (28.3) and *HellaSwag* (30.0), it lags larger models, reflecting limits in multi-step reasoning. *CommonsenseQA* (24.6) and *MMLU* (27.9) remain challenging due to restricted factual coverage. Still, BitMar achieves non-trivial accuracy across all tasks, confirming that extreme compression can yield usable models for targeted workloads, though with expected trade-offs in knowledge-heavy benchmarks.

As shown in Table 2, BitMar achieves an average 60.5% across finetuned NLP benchmarks, with strong results on paraphrase (*QQP*: 70.2%, *MRPC*: 69.1%) and reading comprehension (*BoolQ*: 66.5%), but weaker performance on inference (*MNLI*: 42.3%, *RTE*: 54.0%). Multimodal tasks yield modest scores (21–25%), with the best results on *EWoK* (24.9%), likely benefiting from episodic memory. Linguistic analysis shows reasonable syntax (*BLIMP*: 48.7%) and compositional reasoning (51.5%), but poor morphological productivity (*WUG*: -0.16/-0.22). Overall, BitMar balances extreme efficiency with usable performance, excelling in lightweight reasoning while struggling on complex multimodal and morphological tasks.

5.2 Ablation Study: Episodic Memory

Evaluated under BabyLM 2025 evaluation pipeline (same as Table 2).

Efficiency. As Table 3 reports, a fixed retrieved vector supplies context each step, reducing long-range attention while keeping 1.58-bit compute.

Zero-shot accuracy in Δ (pp). Table 4 reports the performance differences on zero-shot tasks. Overall, the results suggest that incorporating additional contextual information generally enhances task accuracy.

Regressions. We observe two regressions. First, regarding *WUG* morphology, correlations are negative, -0.36 for adjectives and -0.16 for past tense, indicating reduced morphological productivity under extreme quantization. Second, *reading* alignment scores are lower with memory (0.44/0.11) than without (1.11/0.66), suggesting that episodic conditioning can dampen psycholinguistic alignment. Tuning memory capacity or injection strategy may mitigate this.

Fine-tuning. No significant changes on *BoolQ/MNLI/MRPC/MultiRC/QQP/RTE/WSC*, suggesting memory mainly affects generation, not supervised heads.

Model	Native 1-bit	ARC-Easy	BoolQ	HellaSwag	WG	CQA	MMLU
Bonsai 0.5B	✓	58.25	58.44	48.01	54.46	18.43	25.74
OLMo-BitNet 1B	✓	25.38	52.48	25.88	51.54	19.49	25.47
Falcon3-1.58bit 7B	×	65.03	72.14	59.46	60.14	67.08	42.79
LLaMA3-8B-1.58 8B	×	70.71	68.38	68.56	60.93	28.50	35.04
BitNet b1.58 2B	✓	74.79	80.18	68.44	71.90	71.58	53.17
BitMar-14M (Ours)	✓	28.32	42.83	30.04	54.57	24.57	27.90

Table 1: **Benchmark performance on language understanding tasks.** A ✓indicates models trained natively with 1-bit precision. All reported values correspond to task accuracy (%), illustrating BitMar's competitive performance under extreme compression. [WG-WinoGrande; CQA-CommonsenseQA]

Category	Task	Primary Metric	Score
Finetune NLP	BoolQ	Accuracy	66.5%
	MNLI	Accuracy	42.3%
	MRPC	Accuracy	69.1%
	MultiRC	Accuracy	57.6%
	QQP	Accuracy	70.2%
	RTE	Accuracy	54.0%
	WSC	Accuracy	63.5%
Multimodal	DevBench	Visual Vocab Acc.	21.2%
	VQA	Accuracy	21.4%
	Winoground	Accuracy	23.8%
World Knowledge	EWOK	Accuracy	24.9%
Linguistic	BLIMP	Accuracy	48.7%
Reasoning	Compositional	Accuracy	51.5%
	Entity Tracking	Accuracy	31.2%
Psycholing.	Reading Comp.	Score	0.44
Morphology	Wug Adj.	Corr.	-0.16
	Wug Past	Corr.	-0.22

Table 2: BitMar results on BabyLM evaluation tasks.

Metric	Mem. On	Mem. Off
Throughput (tok/s)	57.3	7.7
Latency/token (ms)	17.3	129.8
Energy (J)	1.90	9.17
RAM (MB)	956	1,076

Table 3: **Inference ablation metrics.** Comparison of throughput, latency, energy consumption, and memory usage.

Task	Δ (pp)
Entity Tracking (Split 1)	+2.9
Entity Tracking (Split 2)	+4.1
COMPS	+3.4
BLiMP	+0.6
VQA	+3.4
EWoK (Split 1)	-1.6
EWoK (Split 2)	+1.0
Winoground	-1.6
DevBench	No effect

Table 4: **Ablation results on episodic memory.** Performance differences (Δ, in percentage points), positive values indicate improvements when memory is enabled.

Ablation Summary. Episodic Memory is $\sim$7.5$\times$ faster, using 79% less energy and 11% less VRAM in our tests. It delivers 3 – 4 percentage point gains on entity/property reasoning and multimodal QA, though morphology and some psycholinguistic alignment metrics can degrade. Overall, combining attention sinks with episodic memory enables efficient long-context use under tight resource budgets.

6 Conclusion

BitMar-14M is a compact 1.58-bit multimodal language model using BitNet quantization, DiNOv2 vision compression, cross-modal fusion, an attention-sink decoder for efficient long-context reasoning, and an external episodic latent memory for deployment on resource-constrained edge devices. With adaptive training, it maintains stable alignment and memory use despite its tiny size. Though less accurate than larger low-bit models on knowledge-heavy tasks, it performs competitively on binary reasoning and coreference, showing that 1.58-bit compression and efficient design can enable multimodal reasoning with drastically reduced compute and storage.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millicah, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, Los Alamitos, CA, USA. IEEE Computer Society.

Antoine Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Sebastian Rutherford, Matthew Botvinick, Jean-Baptiste Sifre, and Stan Clark. 2022. Improving language models by retrieving from trillions of tokens. *International Conference on Machine Learning (ICML)*, 162:2209–2226.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM turns 3: Call for papers for the 2025 BabyLM workshop. In *BabyLM Turns 3: Call for papers for the 2025 BabyLM workshop*, pages 2–3, Suzhou, China.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2021. 8-bit optimizers via block-wise quantization. *CoRR*, abs/2110.02861.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: accurate post-training quantization for generative pre-trained transformers. *CoRR*, abs/2210.17323.

Shaikat Galib, Shanshan Wang, Guanshuo Xu, Pascal Pfeiffer, Ryan Chesler, Mark Landry, and Sri Satish Ambati. 2024. H2ovl-mississippi vision language models technical report.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, et al. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. *arXiv preprint arXiv:2402.14905*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. *ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks*. Curran Associates Inc., Red Hook, NY, USA.

Marcelo G. Mattar and Nathaniel D. Daw. 2022. A neural network model of when to retrieve and encode episodic memories. *eLife*, 11:e74445.

Marcelo G. Mattar and Nathaniel D. Daw. 2024. Toward the emergence of intelligent control: Episodic generalization and optimization. *Open Mind*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748 [cs, stat]*. ArXiv: 1807.03748.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. Dinov2: Learning robust visual features without supervision.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, page 647–664, Berlin, Heidelberg. Springer-Verlag.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Pierre Stock, Armand Joulin, Rémi Gribonval, Benjamin Graham, and Hervé Jégou. 2019. And the bit goes down: Revisiting the quantization of neural networks. *CoRR*, abs/1907.05686.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. 2023. Bitnet: Scaling 1-bit transformers for large language models.

Yuanhan Yao, Qinghao Yu, Ao Zhang, Xiaoyi Wang, Zhiyang Xu, Chendong Yuan, Ying Wang, Yaoyao Liu, Kunchang Wang, Yunhai Yu, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Zhengqing Yuan, Zhaoxu Ren, Lichao Feng, Zhi Zhao, Kai Cui, and Shiliang Jiang. 2024. Tinygpt-v: Efficient multimodal large language model via small backbones. *arXiv preprint arXiv:2312.16862*.

Wei Zhang, Xiaoming Liu, Hao Chen, and Yifan Wang. 2024. Tinyllm: A framework for training and deploying language models at the edge computers. *arXiv preprint arXiv:2412.15304*.

Chenzhuo Zhu, Song Han, Huizi Mao, and William J. Dally. 2016. Trained ternary quantization. *CoRR*, abs/1612.01064.

Exploring smaller batch sizes for a high-performing BabyLM model architecture

Sharid Loáiciga, Eleni Fysikoudi, Asad Sayeed
Department of Philosophy, Linguistics and Theory of Science,
University of Gothenburg
sharid.loaiciga@gu.se, gusfysel@student.gu.se, asad.sayeed@gu.se

Abstract

We explore the conditions under which the highest-performing entry to the BabyLM task in 2023, Every Layer Counts BERT or ELC-BERT, is best-performing given more constrained resources than the original run, with a particular focus on batch size. ELC-BERT's relative success, as an instance of model engineering compared to more cognitively-motivated architectures, could be taken as evidence that the "lowest-hanging" fruit is to be found from non-linguistic machine learning approaches. We find that if we take away the advantage of training time from ELC-BERT, the advantage of the architecture mostly disappears, but some hyperparameter combinations nevertheless differentiate themselves in performance.

1 Introduction

The BabyLM Challenge (Warstadt et al., 2023a; Choshen et al., 2024; Charpentier et al., 2025) has become a shared task-style sandbox where researchers are invited to develop language models trained under developmentally plausible data budgets, simulating the linguistic input available to human children up to the age of 13. By setting small-scale amounts of data, either 10M or 100M words depending on the track, and providing standardized evaluation benchmarks, it aims to promote data-efficient modeling architectures. It also aims to support cognitively plausible approaches to automatic language acquisition. Finally, it is intended to broaden participation in language model research beyond large-scale industrial settings.

In this paper, we report on an expanded exploration of results for the ELC-BERT model (Charpentier and Samuel, 2023), the winning submission to the BabyLM Challenge 2023. This exploration focuses on the `strict-small` track, using the evaluation conditions and tools from the 2023 edition. Specifically, we investigate whether ELC-BERT's performance is primarily driven by computational resources, with a particular focus on batch size.

One of the findings of the 2023 edition of the shared task was that architectural innovations tended to be more successful than approaches inspired by curriculum learning or cognitive principles and the ELC-BERT architecture was at the forefront of that. ELC-BERT modifies the standard BERT architecture by replacing uniform residual connections with a learned weighting mechanism so that each layer selectively combines outputs from all preceding layers. This selective weighting means that the model can prioritize information from the most relevant layers, as opposed to treating all layers equally.

The approach achieved very strong performance, which makes ELC-BERT a great candidate as a base system going forward. However, it is also reported to have been trained for very long time and on a very large compute cluster. Our motivation in this work is primarily to investigate whether comparable performance can be achieved using substantially less computational resources using the ELC-BERT "tweak" to the BERT approach.

Since compute capacity is often a major limitation, both in terms of access and cost, efficient training methods are of particular importance. In this sense, it is worth noting that this year the BabyLM shared task has introduced stricter constraints on model training, limiting the number of epochs and training examples (Charpentier et al., 2025). This is a positive step toward standardizing experimental conditions and enabling more meaningful comparisons across replication studies.

Like all shared task submissions, ELC-BERT was likely developed under time constraints. We recognize that this is not an ideal setting for a comprehensive hyperparameter search. Our work addresses this gap by providing a more thorough investigation.

Our main contributions can be summarized as

Proceedings of the First BabyLM Workshop, pages 155–159
November 8, 2025

follows. In scenarios with more constrained computational resources:

- BLiMP scores are ultimately lower than the original ELC-BERT result.
- MSGS scores are higher than the original ELC-BERT result with a range of above-baseline outcomes on GLUE.
- A batch size of 32 with gradient accumulation of 12 (effective batch size of 384) achieves performance comparable to, or exceeding, that of larger batch sizes among the settings we tested.

All told, our experiments show that while ELC-BERT's original success depended originally in large part on the computational resources given to it, there are, nevertheless, some resource-constrained settings on which it does appear to be able to make performance gains.

2 Related work

Across both the 2023 and 2024 editions of the BabyLM Challenge (Warstadt et al., 2023b; Hu et al., 2024), the main insights remain consistent: language models can achieve good performance under strict data budgets, though substantial architectural innovations tend to yield greater gains than curriculum- or cognition-inspired methods. (In this case, "architectural innovations" are shorthand for alterations to Transformer-based machine learning approaches that are not directly inspired by linguistic or neurocognitive insights.) Efficiency in terms of data is the core objective of the task, but results also suggest a strong correlation between computational resources and performance, revealing a trade-off between data-efficiency goals and the benefits of larger-scale training.

Following these findings and with a focus on human-scale language modeling, Wilcox et al. (2025) examined this trade-off in high-performing systems from the shared task including ELC-BERT and LTG-BERT (Samuel et al., 2023), a closely related architecture from the same group that also performed very well. Importantly, they capped training at 20 epochs to control for the very long training of the original. The original ELC-BERT was trained for 31250 training steps and over 2000 epochs, whereas most other participants reported training for roughly 20 epochs. Wilcox et al. report performance comparable to the original, with only a 2-3 point drop in accuracy for both systems. However, specific results for the `strict-small` track for ELC-BERT are not provided (cf. Table A.2 in Wilcox et al.).

Furthermore, the ELC-BERT batch size for `strict-small` in both the LTG-BERT paper and ELC-BERT paper is 32768 with 128 tokens per sequence, totaling approximately 4M tokens. We estimate that this requires 2048 GB VRAM, equivalent to about 26 NVIDIA A100 GPUs. The batch size for the submitted ELC-BERT for `strict-small` is 8096, which we estimate requires about 6 A100 GPUs. In constrast, Wilcox et al. use a batch size of 2048 (cf. Table A.3) which requires 2 A100 GPUs. Lacking these computational resources, we investigate smaller batch sizes and use gradient accumulation as a means to approximate the bigger sizes. This of course also has an impact on the time that experiments run.

Within this context, we do not attempt a full replication of the experimental setup from the original ELC-BERT paper, as our computing infrastructure does not permit it. We focus instead on exploring the prospects for this type of architecture in more resource-constrained settings. These are arguably more plausible and faithful to BabyLM's attempt to simulate the conditions of human language acquisition *in silico*.

We performed a hyperparameter search within our computational constraints, contributing to transparency and supporting reproducibility. In this paper, we focus on batch size as representing one of the main resource bottlenecks of the ELC-BERT approach. We also contribute to the investigation of whether the original performance is obtained primarily from the "Every Layer Counts" architecture innovation or from the availability of substantial computational resources used to train ELC-BERT.

3 Set up

Most experiments in Table 2 were run on a single node of our computing cluster equipped with four NVIDIA Tesla A100 HGX GPUs (80 GB RAM each) and experiments with batches larger than 506 were run on 2 A100fat GPUs. Fine-tuning was performed on two GeForce RTX 3090 GPUs (24 GB RAM each). Pre-training used the hyperparameters specified in Charpentier and Samuel (2023), reproduced in Table 1. For fine-tuning, all hyperparameters from the official evaluation scripts[1] were

[1]https://github.com/babylm/evaluation-pipeline-2023

Hyperparameter	Submitted model
Number of parameters	24M
Number of layers	12
Hidden size is	384
FF intermediate size	1024
Vocabulary size	6 144
Attention heads	6
Hidden dropout	0.1
Attention dropout	0.1
Training steps	31250
Batch size	8096
Initial Sequence length	128
Warmup ratio	1.6%
Initial learning rate	0.005
Final learning rate	0.005
Learning rate scheduler	cosine
Weight decay	0.4
Layer norm ϵ	1e-7
Optimizer	LAMB
LAMB ϵ	1e-6
LAMB β_1	0.9
LAMB β_2	0.98
Gradient clipping	2.0
Gradient accumulation	1

Table 1: Pre-training hyperparameters for ELC-BERT model trained on the STRICT-SMALL track reported in Charpentier and Samuel (2023)

left unchanged[2].

4 Results and Discussion

Several observations can be made from Table 2. First, BLiMP scores, which focus on fine-grained grammatical knowledge, tend to be much lower in our replications compared to the original ELC-BERT results, even when training for the same number of steps. In contrast, although GLUE scores are also in general lower, the gap is not as wide. MSGS scores, however, always improve. Importantly, we note that GLUE and MSGS are obtained after a fine-tuning stage for which the default hyperparameters from the BabyLM evaluation set-up were used.

[2]In private communication with the original authors, we discovered that they were using an AMD-based architecture. On further investigation, we discovered that there are significant differences in the implementation of synchronization and gradient accumulation between AMD and NVIDIA that may have an effect on results.

A second observation concerns batch size and gradient accumulation. Runs using a batch size of 32 with gradient accumulation of 12 (effective batch size 384) achieves performance on GLUE and MSGS that matches or exceeds that of much larger batch sizes, while requiring significantly fewer computational resources. However, BLiMP performance seems insensitive to this and does not increase.

Training duration also emerges as an important factor. The original ELC-BERT was trained for over 2000 epochs, a scale of computation probably beyond what most academic teams can access. By comparison, our most efficient runs complete in minutes to a few hours, making them feasible for small groups or even individual researchers. This gap raises the question of how much infrastructure is required to remain competitive in modern NLP research and stresses the importance of computational budgets as well as data budgets.

Due to limitations in our computing infrastructure, we were unable to replicate the original batch size of 8096 and more than 2000 epochs. It remains an open question whether extended training can lead to convergence on different optima in the parameter space, given that language modeling does not converge toward a single optimal decision boundary.

5 Conclusions

We have evaluated the performance of ELC-BERT on a constrained setup and across several batch sizes, obtaining results that differ greatly from those reported in the original system description in a manner that suggests that the computational resources are, perhaps unsurprisingly, key to high performance even in data-constrained conditions. This is nevertheless significant because it *could* have been the case that the architectural innovation of ELC-BERT would have an much bigger influence on the outcome even under an environment of restricted computation.

In the BabyLM task, there is a healthy emphasis on comprehensive reporting of experimental conditions, including hyperparameters, training setup, and hardware specifications. Participants in the shared task complete a form where this information is reported. Future work in this area could involve incorporating such information into the benchmark itself, which could strengthen transparency and comparability across submissions.

Pre-training							Fine-tuning	
Batch size	**Training steps**	**Gradient accum.**	**Epochs**	**Time**	**BLiMP**	**BLiMP suppl.**	**GLUE**	**MSGS**
Original								
8096	31250	1	>2000	–	80.00	67.00	73.7	29.4
32	15625	1	4	21m	51.03	47.08	55.93	46.94
32	31250	1	7	44m	50.18	46.89	57.89	43.67
32	15625	12	41	2h39m	50.53	50.70	63.20	43.71
256	15625	1	27	1h7m	44.85	50.59	63.23	43.62
256	31250	1	53	19h57m	50.37	47.07	65.46	39.62
256	125000	1	218	8h31m	44.85	50.59	65.46	39.62
256	250000	1	437	17h4m	44.17	49.49	65.46	39.62
256	15625	12	333	5d10h5m	47.72	49.41	63.66	39.31
512	15625	1	55	1h49m	50.04	46.94	62.38	43.17
512	31250	1	109	3h37m	52.22	45.65	63.80	43.15
253	31250	32	1479	5d22h29m	46.95	49.88	63.72	39.31
506	31250	16	1736	3d18h42m	49.03	49.36	63.64	39.31

Table 2: ELC-BERT re-runs with varying batch sizes. Reported scores are average accuracies.

The shared task is still young, and beyond the work of Wilcox et al., there are very few analyses of this kind. We believe that such investigations are important for understanding both the reproducibility aspects and the broader implications of results in this setting.

A note on reproducibility The reproducibility crisis has been a subject of discussion for several years in addition to the usual pressures of the academic publishing cycle[3] (Baker, 2016), and computational linguistics and NLP are no exception. Despite its central role in scientific progress, reproducibility remains a persistent challenge in NLP research (Belz, 2022). In response, since 2020, reproducibility checklists are a requirement at submission time (Dodge et al., 2019; Magnusson et al., 2023), and initiatives such as ReproNLP, a shared task on reproducibility, have emerged (Belz et al., 2025). Nonetheless, the practical difficulties of reproducing scientific work are something that nearly every researcher eventually encounters.

In a shared task such as BabyLM, there is a risk that the whole task "overfits" on the most successful-seeming approaches on a year-to-year basis, in an environment in which the problem space is still not well defined (i.e., what even *is* an appropriate measure of human acquisition realism in language modeling?). Therefore, we argue that both replications *and* hyperparameter searches are core tasks in the BabyLM context, especially since the latter ensures that a result is placed in its proper theoretical context.

[3] https://www.nature.com/articles/d41586-024-04253-w

Limitations

Our computing infrastructure does not permit a direct replication of the original ELC-BERT training conditions. This limitation means that our work does not address reproducibility in the strict sense. Instead, it evaluates whether the original findings hold when training is conducted under scaled-down computational settings, and our conclusions should be interpreted within this narrower scope. Furthermore, no learning rate adjustments were made in conjunction with the smaller batch sizes, which may introduce bias into the results.

Acknowledgments

The work reported in this paper has been supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg. Addi-

tional funding was provided by the Gothenburg Research Initiative for Politically Emergent Systems (GRIPES) supported by the Marianne and Marcus Wallenberg Foundation grant 2019.0214.

The computations and data storage were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

The authors gratefully acknowledge the support provided by the developers of the ELC-BERT system and the organizers of BabyLM, particularly for their assistance with technical issues and specific inquiries.

References

Monya Baker. 2016. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454.

Anya Belz. 2022. A Metrological Perspective on Reproducibility in NLP*. *Computational Linguistics*, 48(4):1125–1135.

Anya Belz, Craig Thomson, Javier González Corbelle, and Malo Ruelle. 2025. The 2025 ReproNLP shared task on reproducibility of evaluations in NLP: Overview and results. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 1002–1016, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candance Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Ian Magnusson, Noah A. Smith, and Jesse Dodge. 2023. Reproducibility in NLP: What have we learned from the checklist? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12789–12811, Toronto, Canada. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Adina Williams, Bhargavi Paranjabe, Tal Linzen, and Ryan Cotterell. 2023b. Findings of the 2023 BabyLM Challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the 2023 BabyLM Challenge*. Association for Computational Linguistics (ACL).

Ethan Gotlieb Wilcox, Michael Y. Hu, Aaron Mueller, Alex Warstadt, Leshem Choshen, Chengxu Zhuang, Adina Williams, Ryan Cotterell, and Tal Linzen. 2025. Bigger is not always better: The importance of human-scale language modeling for psycholinguistics. *Journal of Memory and Language*, 144:104650.

BLiSS 1.0: Evaluating Bilingual Learner Competence in Second Language Small Language Models

Yuan Gao* **Suchir Salhan*** **Andrew Caines**
Paula Buttery **Weiwei Sun**
ALTA Institute Department of Computer Science & Technology, University of Cambridge

Abstract

To bridge the gap between performance-oriented benchmarks and the evaluation of cognitively-inspired models, we introduce **BLiSS 1.0**, a Benchmark of Learner Interlingual Syntactic Structure. Our benchmark operationalizes a new paradigm of selective tolerance, testing if a model finds a naturalistic learner error more plausible than a matched, artificial error within the same sentence. Constructed from over 2.8 million naturalistic learner sentences, BLiSS provides 136,867 controlled triplets (corrected, learner, artificial) for this purpose. Experiments on a diverse suite of models demonstrate that selective tolerance is a distinct capability from standard grammaticality, with performance clustering strongly by training paradigm. This validates BLiSS as a robust tool for measuring how different training objectives impact a model's alignment with the systematic patterns of human language acquisition.

BLiSS on HuggingFace (BLiSS 1.0 Dataset and Pretrained Models)

Training Code Open-Sourced on GitHub

1 Introduction

There is a growing interest in the NLP community in developing models that are not just powerful, but also cognitively inspired—that is, models which aim to reflect the processes of human language acquisition. Current evaluation benchmarks for language models are overwhelmingly performance-oriented, centering around grammaticality tests, adherence to standard grammar, and task performance (e.g., BLiMP Warstadt et al. (2020) and GLUE Wang et al. (2018)). While these measures are informative in evaluating linguistic competence, the core question for cognitively inspired modeling is different: do our systems exhibit the kinds of behaviors that emerge in human acquisition? For models that aim to be cognitively plausible, we need a complementary, acquisition-focused perspective, one that inspects how grammar competence is organized and learned.

This evaluation gap is particularly important for models of Second Language Acquisition (SLA), which we refer to as L2LMs (Aoyama and Schneider, 2024). A central characteristic of the SLA process is the production of systematic 'errors'. These deviations are not random noise, but rather structured evidence of the learner's developing internal grammar, or "interlanguage" (Corder, 2015; Selinker, 1972). For a model to be truly 'learner-like', it must be sensitive to these specific, structured patterns observed in real human data.

To address this, we propose a new paradigm built on a key assumption: the systematicity of learner errors is tied to both the type of error and its specific locus within a sentence. This assumption, therefore, theorizes that moving an attested error to a different, albeit plausible, location renders it less naturalistic and less human-like. This approach, which uses an error's locus as a test of naturalness, is inspired by similar methodologies for evaluating complex linguistic phenomena (Sterner and Teufel, 2025). This allows us to test a model's selective tolerance: its ability to penalize a naturalistic human error less severely than a contrived artificial-locus error of the same type.

We introduce the **Benchmark of Learner Interlingual Syntactic Structure (BLiSS 1.0)**, a large-scale evaluation dataset on a model's alignment with naturalistic language learner patterns, offering a new dimension of evaluating acquisition-focused models. BLiSS is built upon three of the largest English learner corpora available: the EF-Cambridge Open Language Database (EFCAMDAT) (Geertzen et al., 2014), the Write & Improve Corpus (W&I) (Nicholls et al., 2024), and

*Corresponding Authors: yg386@cam.ac.uk, sas245@cam.ac.uk

Proceedings of the First BabyLM Workshop, pages 160–174
November 8, 2025

the First Certificate in English (FCE) dataset (Yannakoudakis et al., 2011).

(1) U:DET (Unnecessary determiner)

a. There are a lot of benefits when we play sports.

b. *There are a lot of benefits when we play **the** sports.

c. **There are a lot of benefits when **the** we play sports.

The core of the BLiSS evaluation is the triplet, a controlled comparison between: a corrected sentence, the original sentence with one error from a learner, and a version with an artificially-generated error of the same error type, as shown in (1). From an initial pool of over 2.8 million raw learner sentence-corrected sentence pairs, we systematically generate a matched artificial-locus for each valid, single-edit grammatical deviation. After a rigorous multi-stage validation pipeline, BLiSS comprises 136,867 high-quality evaluation triplets. Each triplet is accompanied by rich metadata, including learner L1, proficiency level, and error type, as illustrated in Figure 1.

In this paper, we deploy BLiSS to evaluate a diverse suite of models, from large bilingual LLMs to acquisition-inspired L2LMs. Our results yield two key findings. First, we demonstrate that selective tolerance is a distinct capability from standard grammaticality; high performance on BLiMP does not guarantee high performance on BLiSS. Second, we show that model performance on BLiSS clusters strongly by training paradigm, validating it as a tool for measuring how different architectures and training objectives impact a model's alignment with the systematic patterns of human learner language.

2 Related Work

2.1 Second Language Acquisition-Inspired Language Models (L2LMs)

We use L2LMs to denote cognitively inspired models of L2 acquisition (Aoyama and Schneider, 2024). Early work examined transfer—training on an L1 then an L2—and the role of typological distance (Yadavalli et al., 2023; Oba et al., 2023), while later studies add cognitive priors (e.g., alignment to learner reading times; preserving L1 knowledge to probe the Critical Period Hypothesis) and compare sequential vs. mixed L1/L2 exposure (Aoyama and Schneider, 2024; Clahsen and Felser, 2006; Constantinescu et al.; Lenneberg, 1967; Kirkpatrick et al., 2017; Arnett et al., 2025). Given heterogeneous architectures and pretraining corpora (from learner-like data to web-scale sources such as CC-100; Wenzek et al., 2020), a common benchmark tied to learner behavior is needed (Salhan et al., 2024; Arnett et al., 2025).

```
{
  "learnerID": "8421",
  "L1": "Vietnamese",
  "cefr": "C1",
  "topic": "play  sports ",
  "corrected": "There are a lot of
      benefits when we play sports .",
  "learner error": "There are a lot of
       benefits when we play the
      sports .",
  "artificial error": "There are a lot
       of benefits when the we play
      sports .",

  "errant_edits": [{
    "type": "U:DET",
    "o_str": "the",
    "c_str": ""
  }],
  "all_error_types": [
    "U:DET"
  ]
}
```

Figure 1: An example BLiSS triplet illustrating an Unnecessary Determiner (U:DET) error. The original learner sentence contains an unnecessary determiner "the", which is removed in the corrected sentence. Artificially-generated errors of the same type allow controlled evaluation of model preferences.

2.2 Learner Corpora and Error Profiling

Large-scale learner corpora provide an important empirical basis for modeling and evaluating L2 learner behavior. The Write & Improve Corpus 2024 (Nicholls et al., 2024) contains learner essays with Common European Framework of Reference for Languages (CEFR) annotations and corresponding error-labeled corrections. The essays were submitted by users of the 'Write & Improve' writing practice platform[1]. W&I uses ERRANT (Bryant et al., 2017) to annotate errors in learner essays automatically. ERRANT annotations classify errors as replacements (R), missing (M) or unnecessary (U) and assign a specific tag (e.g., M:ADJ means the text omits an adjective). A full table of ERRANT error codes are included in *Appendix C* for reference. The EF-Cambridge Open Lan-

[1] https://writeandimprove.com/

guage Database (EFCAMDAT) (Geertzen et al., 2014) offers a large collection of learner texts annotated with proficiency levels and metadata on learner nationality. Note that the proficiency levels in EFCAMDAT relate to difficulty level attained by users of the 'EF Englishtown' platform (now 'EF English Live'[2]), rather than human ratings of the texts themselves, but this information serves as a good proxy for learner proficiency. The W&I-2024 corpus has a wider range of L1s compared to other publically-available learner corpora, like the FCE subset of the Cambridge Learner Corpus (Yannakoudakis et al., 2011). There are other error-annotated English learner corpora, such as NUCLE (Dahlmeier et al., 2013), JFLEG (Napoles et al., 2017) and Lang-8 (Mizumoto et al., 2012; Tajiri et al., 2012), but are respectively age/language restricted; use fluency rewrite rather than minimal grammatical edits; and have user-generated corrections (Nicholls et al., 2024).

3 BLiSS 1.0

3.1 Motivation

The BLiSS 1.0 benchmark is a large-scale evaluation suite composed of controlled triplets designed to test a model's selective tolerance for naturalistic learner production errors. The evaluation framework for BLiSS is designed to move beyond evaluations of the formal competence of a Language Model (e.g., using broad-coverage datasets like BLiMP (Warstadt et al., 2020)) to evaluate the **alignment of a language model with second language acquisition**. BLiSS builds upon previous attempts to extend acquisition-inspired evaluation frameworks for Language Models (e.g., Evanson et al. (2023)) beyond first language acquisition.

BLiSS 1.0 focuses on naturalistic production errors in learner corpora. The BLiSS 1.0 benchmark is designed to evaluate how closely a language model's outputs align with patterns observed in second language (L2) learners, particularly in terms of grammatical errors. While it is true that individual learner errors do not imply that a majority of learners would make the same mistake in a given sentence, BLiSS focuses on systematic tendencies in learner language rather than absolute probabilities of specific errors. By aggregating errors across millions of sentence-correction pairs from multiple learner corpora, BLiSS captures the distributional patterns of learner errors that are prevalent in naturalistic L2 production. BLiSS does not encourage models to prefer errors, but rather tests alignment with learner error patterns.

This approach addresses a critical limitation of traditional LM evaluation benchmarks (e.g., BLiMP), which primarily assess formal grammatical competence. Such benchmarks assume that the model should always prefer grammatical sentences, but human learners – especially in L2 acquisition –frequently produce systematic errors that reveal underlying acquisition stages, interlanguage phenomena, or L1 transfer effects. BLiSS thus extends evaluation beyond formal competence, providing a framework to test whether models *selectively tolerate or reproduce error patterns* in ways that resemble human learners.

Concretely, we develop BLiSS to enable the study of:

1. **Error-type sensitivity:** Whether language models recognize and react differently to common L2 errors (e.g., determiner omission, verb tense errors).

2. **Position awareness:** By generating artificial errors at positions distinct from the learner's original error, we can test if language models are sensitive to the locus of grammatical deviations, not just their existence.

3. **Learner-informed evaluation:** Leveraging metadata such as L1 background and proficiency level that are available in large-scale corpora allows analysis of model behavior in the context of typologically diverse learner populations.

While BLiSS does not imply that *all learners* would produce a given error, it provides a systematically sampled and validated set of errors that represents frequent phenomena in learner production (Alexopoulou et al., 2015; Le Bruyn and Paquot, 2021; Crossley and Kyle, 2022; Alexopoulou et al., 2022). This makes BLiSS a meaningful benchmark for probing the alignment of language models with human L2 acquisition patterns, without conflating individual idiosyncrasies with population-level tendencies.

3.2 Data: Source Corpora

The credibility and naturalistic grounding of the BLiSS benchmark stem from its foundation in

[2] https://englishlive.ef.com/

large-scale, naturalistic learner data. We aggregate sentence-correction pairs from three of the most widely-used English learner corpora, ensuring our benchmark reflects genuine learner behaviour in communicative contexts.

- The EF-Cambridge Open Language Database (EFCAMDAT) (Geertzen et al., 2014): A very large collection of over 1 million learner texts from an online English learning platform. Texts are annotated with metadata including learner nationality and proficiency levels mapped to the Common European Framework of Reference for Languages (CEFR).

- The Write & Improve (W&I) Corpus (Nicholls et al., 2024): A dataset of learner essays submitted to an online writing feedback tool. It is richly annotated with CEFR levels (A1–C2) and explicit learner L1 labels, providing high-quality metadata for fine-grained analysis.

- The First Certificate in English (FCE) Dataset (Yannakoudakis et al., 2011): A well-known subset of the Cambridge Learner Corpus containing essays from an official language proficiency exam. This provides a valuable sample of argumentative, exam-style writing from a diverse set of L1 backgrounds.

Collectively, these corpora provide a massive pool of over 2.8 million raw sentence-correction pairs, forming the empirical starting point for our triplet construction pipeline, detailed in the following section.

Corpus	# Raw Pairs
EFCAMDAT	2,711,188
W&I	63,926
FCE	52,421
Total	2,827,535

Table 1: Summary of raw single-edit sentence-correction pairs from the source corpora.

3.3 Triplet Construction Pipeline

The construction of the BLiSS dataset follows a multi-stage pipeline designed to transform raw sentence-correction pairs from the source corpora into high-quality, validated triplets. The pipeline emphasizes grammatical precision, methodological transparency, and the atomization of errors to ensure each triplet tests a single distinct linguistic phenomenon.

Grammatical Error Classification and Filtering The process begins with a comprehensive error analysis of the raw sentence pairs using the ERRANT toolkit (Bryant et al., 2017). With these annotations, we first filtered out pairs containing only non-grammatical edits, such as spelling, punctuation, or capitalization changes.

Error Atomization We then atomized sentence pairs with multiple corrections using the ERRANT annotations as a guide. Each distinct grammatical edit within a multi-error sentence was isolated to create a new single-edit pair consisting of the corrected sentence and a version with just that one specific error. This process ensures that every triplet in the final dataset is anchored to exactly one grammatical deviation, allowing for a clean and targeted evaluation.

Rule-Based Artificial Error Generation The core of the pipeline is the generation of an artificial error for each single-edit pair. This rule-based system uses linguistic analysis and morphological generation, creating a new sentence that adheres to two fundamental constraints:

1. **Error Type Consistency:** the artificial error must mirror the grammatical operation of the human error. For example, a missing determiner (M:DET) in the learner sentence prompts the generation of a new sentence where a determiner is removed.

2. **Position Divergence:** The artificial error must be introduced at a different word position than the learner error. This ensures the model is being tested on its sensitivity to the error's locus, not merely its presence.

Multi-Stage Quality Validation To ensure the integrity of BLiSS, every generated triplet was subjected to a rigorous multi-stage validation filter. A triplet was only retained if it passed all of the following checks:

1. **Morphological Correctness:** All inflected words (e.g., verbs, nouns) generated by LemmInflect[3] must be valid English forms.

[3] https://github.com/bjascob/LemmInflect

2. **Triplet Uniqueness:** The artificial error sentence must be distinct from both the corrected sentence and the original learner error sentence.

3. **Error Type Confirmation:** Finally, we used ERRANT as a verifier. The generated artificial error, when compared to the corrected sentence, must be classified by ERRANT as having the same error type as the original human error.

This stringent validation process resulted in an overall success rate of 4.8%, yielding a final dataset of 136,867 high-quality triplets. The low success rate is a direct reflection of the strictness of our quality controls, ensuring that every item in BLiSS is a valid and non-ambiguous test case. A sample of 100 triplets was also manually reviewed, confirming a grammatical and positional accuracy rate of over 95%.

3.4 Dataset Composition

Following the rigorous construction and validation pipeline, the final BLiSS dataset comprises 136,867 high-quality triplets. The composition of the dataset reflects both the diversity of the source corpora and the targeted nature of our filtering process. As shown in Table 2, the majority of the final dataset (76.7%) is derived from the large-scale EFCAMDAT corpus, supplemented by high-quality and diverse data from the W&I and FCE corpora.

Corpus	Triplets	Percentage
EFCamDat	105,034	76.7%
Write & Improve	17,380	12.7%
FCE	14,453	10.6%
Total	136,867	100%

Table 2: BLiSS Composition

Error Type Distribution The dataset provides robust coverage across a range of core grammatical error categories that are common in second language acquisition. Table 3 details the distribution of the five most frequent error types, which collectively account for over 67% of the dataset.

Learner Demographics The rich metadata from the source corpora allows for detailed analysis across learner populations. Table 4 shows the distribution of the top five L1 backgrounds in the dataset.

Error Type	Count	Percentage
M:DET	26,008	19.0%
R:NOUN:NUM	21,149	15.5%
R:PREP	18,702	13.7%
U:DET	15,708	11.5%
R:VERB:TENSE	10,599	7.7%

Table 3: Distribution of the top 5 ERRANT error types in BLiSS.

The significant representation of typologically diverse languages such as Chinese, Japanese, and Arabic makes the benchmark particularly powerful for investigating L1 transfer effects.

L1 Background	Count	Percentage
Chinese	23,771	17.4%
Japanese	14,478	10.6%
Italian	11,918	8.7%
French	11,486	8.4%
Arabic	9,484	6.9%

Table 4: Distribution of the top 5 L1 backgrounds in BLiSS.

In terms of learner proficiency, BLiSS spans a wide range of the CEFR scale, from beginner (A1) to advanced (C2), as detailed in Figure 2. The dataset has substantial representation between the beginner and intermediate(A1 - B2) levels but significantly less at higher levels with only 25 triplets at the C2 level. This broad distribution is a key strength, enabling the study of how model behavior might differ when evaluated on errors typical of different proficiency levels.

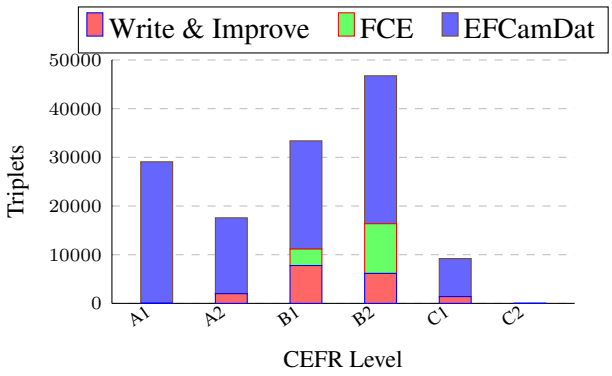


Figure 2: Distribution of CEFR proficiency levels in BLiSS by corpus (stacked triplet counts).

4 Evaluation

The core objective is to quantitatively measure a model's alignment to the naturalistic production errors produced by second language (L2) learners of English. To evaluate a model's selective tolerance, we introduce a set of complementary metrics that capture different aspects of its behavior. The **Learner Preference (LP)** metric provides a simple metric that measures whether the model prefers a human learner sentence over the corrected version, though a high LP could reflect either accurate simulation of learner tendencies or poor grammatical knowledge. To directly probe selective tolerance, **Human vs. Artificial Preference (HAP)** measures whether the model favors naturalistic learner errors over contrived, artificial errors, while **HAP-τ** is a stricter version that ensures the model's preference is meaningful and not just due to numerical noise. Finally, the **Strict Order (SO)** metric captures the most stringent behavior, requiring the model to rank all three sentences in the hypothesized order—corrected first, learner second, artificial last—indicating a balance between grammatical competence and nuanced sensitivity to L2 error patterns. Together, these metrics provide a multi-faceted view of whether a language model can recognize correct grammar, differentiates between plausible and implausible errors, and exhibits robust, cognitively plausible error sensitivity.

A model's preference for a sentence is quantified using token-normalized surprisal, measured in Bits Per Token (BPT), where low BPT indicates high plausibility under the model's learned distribution and high BPT signals a grammatical deviation. By computing BPT scores for each sentence in a BLiSS triplet—including the corrected sentence, the human learner error, and an artificially generated error—we can evaluate not only whether a model recognizes correct grammar, but also whether it differentiates between naturalistic learner errors and contrived mistakes. These BPT scores underpin the three evaluation metrics in the BLiSS framework.

We recommend that **each metric should be reported separately**, as they provide complementary insights: LEARNER PREFERENCE (LP) captures general grammatical preference, HUMAN V ARTIFICIAL PREFERENCE (HAP and HAP-τ) metrics assess selective tolerance, and STRICT ORDER (SO) evaluates the full hypothesized ranking. Combining these metrics into a single score would obscure these distinctions and reduce the interpretability of a language model's behavior on L2 error patterns.

4.1 Scoring Signal

We quantify a model's preference for a given sentence s by its token-normalized surprisal, measured in Bits Per Token (BPT). This is calculated as the negative log-likelihood of the sentence, normalized by the number of tokens.

$$BPT(s) = -\frac{1}{|s|}\sum_{t=1}^{|s|} log_2 p(w_t|w_{<t})$$

where $|s|$ is the number of tokens in the sentence and $p(w_t|w_{<t})$ is the probability assigned by the model to token w_t given the proceeding context.

From a cognitive perspective, surprisal is often used as a proxy for processing effort. A sentence that aligns with a model's learned grammatical and statistical patterns will have low surprisal (low BPT), indicating it is highly plausible under the model's distribution. Conversely, a sentence with a grammatical deviation will have high surprisal (high BPT). This allows us to use BPT as a 'plausibility score' to measure the model's preference for each of the three sentences in a BLiSS triplet.

For each item in the BLiSS dataset, we apply this scoring signal to three sentences in the triplet, which we will formally denote as: s_{corr} (corrected), s_{lrn} (learner), and s_{art} (artificial).

4.2 Evaluation Metrics

We present a suite of metrics designed to provide a multi-faceted view of a model's behavior. These metrics are organized around two key concepts: a baseline measure of simple learner preference and our primary measures of selective tolerance.

Baseline Metric: Learner Preference (LP) We provide this metric as a minimal-pair evaluation. We define Learner Preference (LP) as the proportion of items where the model prefers the learner sentence over the corrected version: $BPT(s_{lrn}) < BPT(s_{corr})$. The motivation for LP is that for certain applications, such as simulating learner output, a model might be intentionally designed to reproduce learner errors. However, LP is inherently ambiguous, as a high score could also simply reflect poor grammatical knowledge. We therefore use it as a diagnostic baseline.

Selective Tolerance Metrics To overcome the ambiguity of LP, our primary metrics are designed to probe a model's selective tolerance directly. The desired behavior for a cognitively plausible model is twofold: it should, first and foremost, still recognize and prefer correct grammar, yet it should also differentiate between the plausibility of different types of errors. Specifically, it should find a naturalistic, systematic learner error to be more plausible (less surprising) than a contrived, artificial error. According to this principle, the ideal ordering of preferences for any triplet should be the corrected sentence, followed by the learner sentence, and finally the artificial sentence. Following this, we present three primary metrics that quantify a model's adherence to this behavior.

1. **SO (Strict Order):** This is the most stringent metric. It measures the proportion of the items where the model's preferences follow the full, hypothesized order of plausibility: $BPT(s_{corr}) < BPT(s_{lrn}) < BPT(s_{art})$. A high SO score is the strongest evidence that a model successfully balances grammatical competence with a nuanced sensitivity to interlanguage.

2. **HAP (Human vs. Artificial Preference):** This metric isolates the central test of selective tolerance by measuring the proportion of items where the model simply prefers the human error over the artificial one: $BPT(s_{\text{lrn}}) < BPT(s_{\text{art}})$. HAP allows us to credit a model for correctly distinguishing between the two error types.

3. **HAP-τ (Robust HAP):** A stricter version of HAP, this metric requires the BPT difference between the artificial and learner sentences to exceed a small positive buffer τ: $BPT(s_{\text{art}}) - BPT(s_{\text{lrn}}) > \tau$. This ensures the model's preference is confident and meaningful, rather than an artifact of numerical noise.

5 Models

We evaluate a diverse range of models on the BLiSS benchmark. The models are grouped into four distinct families, ordered by their increasing degree of specialization for second language acquisition (SLA). This progression allows us to systematically investigate how training data, architecture, and SLA-inspired objectives influence a model's capacity for selective tolerance.

Standard Bilingual LLMs This family serves as our baseline, representing powerful, general-purpose models that have not been specifically designed to model learner language or the acquisition process. These are large language models trained on massive corpora of standard, native-speaker text in two languages. Their training objective is to model fluent, grammatical language, not the intermediate stages of learning. We include Bilingual-GPT-NeoX-4B[4] (Japanese–English) (Zhao et al.; Sawada et al., 2024), CroissantLLM[5] (Faysse et al., 2024) (French–English), and MAP-Neo-7B[6] (Zhang et al., 2024) (Chinese–English).

Bilingual BabyLMs This family represents models that are 'acquisition-inspired' in their data scale but are not explicitly designed for SLA. These are smaller models trained from scratch on developmentally plausible, child-directed speech (CDS) in two languages. While they model the acquisition of language, they are primarily simultaneous bilingual first language acquisition (BFLA), not successive L2 learning. We evaluate publicly released models (Jumelet et al., forthcoming)[7] trained from scratch on 10M words of CDS in English plus one other language (Persian, German, Indonesian, Japanese, Dutch, or Chinese).

Acquisition-Inspired L2 Models This family includes models that explicitly incorporate principles from SLA research into their design. They are designed to simulate the process of an L1 speaker learning an L2, often through sequential training regimes or other architectural priors that model transfer. SLABERT (Yadavalli et al., 2023) follows the *Test for Inductive Bias via Language Model Transfer* (TILT; Pauls and Klein, 2012): pretrain on age-ordered CDS in L1 (French, Polish, Indonesian, Japanese), then fine-tune on English adult-directed speech with all parameters frozen except embeddings. B-GPT (Arnett et al., 2025) is trained with *sequential* exposure (L1 then L2) or *simultaneous* exposure (L1+L2 mixed), here evaluated only for English L2 with Dutch or Spanish L1.

Learner-Trained Models This final family represents models that are directly exposed to learner language during training. Instead of learning from

[4] https://huggingface.co/rinna/bilingual-gpt-neox-4b

[5] https://huggingface.co/croissantllm/CroissantLLMBase

[6] https://map-neo.github.io/

[7] https://huggingface.co/BabyLM-community

native text and hoping learner-like patterns emerge, we train these models on the same kind of data used in our benchmark. We train several GPT-2 medium models from scratch on learner-produced English essays from the Cambridge Learner Corpus (CLC) and EFCAMDAT. To ensure fairness in evaluation, these models are only evaluated on the W&I slice of BLiSS.

6 Results

Table 5 presents BLiSS scores for all evaluated models. Our analysis shows three primary findings that validate the BLiSS benchmark as a tool for measuring a distinct, acquisition-related dimension of model behavior.

An analysis of the model families in Table 5 reveals distinct performance profiles. The Bilingual LLMs and B-GPT models emerge as the strongest performers on our primary selective tolerance metrics. Both families form tight clusters with high HAP scores (≈66-67%) and, notably, the highest Strict Order (SO) scores (≈55-57%). This indicates a robust ability to correctly rank the full triplet.

The Bilingual BabyLM models also perform significantly above chance, but with lower SO scores (≈35-44%), suggesting a weaker, though still present, signal of selective tolerance. A consistent and important trend among these three families is a statistically significant increase in performance on their respective L1 data slices, providing strong evidence that they have internalized L1-dependent transfer patterns and validating BLiSS as a tool for probing these fine-grained behaviors.

In contrast, the SLABERT and Learner-Trained models show a different and less successful profile. Their very high Learner Preference (LP) scores (often >50%) are coupled with poor performance on our primary selective tolerance metrics, particularly Strict Order. This suggests that their training may have made them indiscriminately accepting of learner-like forms, hindering their ability to distinguish between plausible human errors and implausible artificial ones.

6.1 BLiSS vs. BLiMP

To visualize the relationship between a model's BLiSS and BLiMP scores, Figure 3 plots the HAP score against the BLiMP score for each evaluated mode, colour-coded by the model family. The plot demonstrates several key insights into the nature of the BLiSS benchmark and the capabilities of

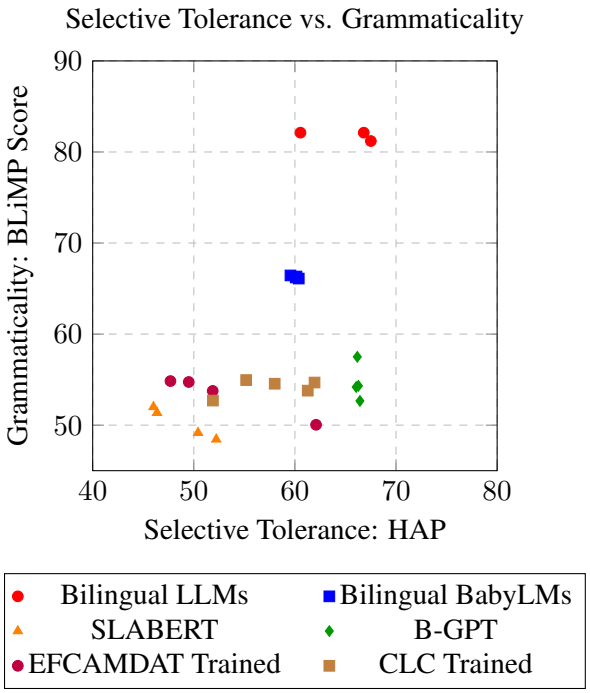


Figure 3: Selective tolerance (HAP score) versus grammaticality (BLiMP score) across all evaluated models. Each point represents a model, colour-coded by its training family.

different model architectures.

A striking observation is that models from the same training family form tight clusters. For example, the large Bilingual LLMs occupy a distinct region in the top-right of the plot, while the B-GPT and SLABERT models form their own clear groups. This consistency is a powerful validation of our methodology; it suggests that BLiSS is successfully capturing a stable signal that is reflective of the underlying training paradigm, rather than just idiosyncratic model behavior. The two learner-trained families (EFCAMDAT and CLC) show slightly more internal variance, which is expected, as the primary differentiating factor within those families is the training data.

Another pattern we observe from the plot is the clear lack of a strong positive correlation between the two metrics. High performance on BLiMP does not guarantee high performance on BLiSS and vice-versa. The large Bilingual LLMs, for instance, excel at both. However, other models achieve strong selective tolerance without top-tier grammaticality. The B-GPT models are a prime example.

This demonstrates that BLiSS offers a complementary, second dimension for language model evaluation. It measures a distinct capability that

Model	BLiSS							BLiMP
	HAP		HAP@τ		SO		LP	
	Overall	L1	Overall	L1	Overall	L1		
Bilingual LLMs								
CroissantLLM	67.51*	81.76**	57.42*	71.70**	57.64*	54.09	12.84	81.20
Neox-4B	60.55*	78.26**	42.98*	62.32**	35.15*	1.88	16.27	82.12
MAP-Neo-7B	66.81*	77.12	58.14*	72.03**	56.05*	45.76**	14.14	82.12
Bilingual BabyLM models								
BBLM–DE	60.15*	76.92**	50.59*	66.67**	43.73*	33.33	18.80	66.32
BBLM–ZH	59.56*	72.88**	49.57*	66.95**	34.93*	38.14	20.44	66.44
BBLM–ID	60.08*	66.6	50.41*	62.96	43.66*	37.04	28.57	66.22
BBLM–FR	60.38*	79.25**	50.70*	67.92**	43.93*	44.03	13.71	66.10
SLABERT								
SLABERT–JP	50.42*	47.83	31.40*	27.54	16.58*	15.94	63.46	49.16
SLABERT–FR	52.22*	52.20	34.50*	33.96	16.20*	15.09	57.47	48.44
SLABERT–ID	46.36*	38.89	30.91*	25.93	15.40*	12.96	57.14	51.36
SLABERT–PL	46.01*	54.92**	32.99*	38.52	14.07*	18.85	57.57	52.00
B-GPT								
B-GPT-ES-SIM	66.43*	77.14**	56.48 *	62.14**	54.57*	50.00	12.99	52.66
B-GPT-ES-SEQ	66.06*	74.29**	56.15*	61.43**	55.06	48.57**	12.17	54.19
EFCAMDAT Trained								
LM–EF	51.87*	47.94**	36.94*	33.78**	15.23*	12.66**	39.51	53.76
Noise–EF	47.69*	46.26	28.47*	29.67	11.40*	11.33	41.95	54.84
Contr–EF	62.09*	62.24	48.02*	41.74	23.70*	18.03	69.51	50.04
Compl–EF	49.50*	56.23	44.75*	45.80	21.35*	19.54	40.73	54.74

Table 5: BLiSS metrics (HAP, HAP@τ, Strict Order, LP) with L1 and Overall subcolumns, alongside BLiMP grammaticality accuracy. An asterisk (*) indicates performance significantly above the 50% chance baseline ($p < 0.05$), while a double asterisk (**) on L1 scores indicates a statistically significant difference between the L1-specific and overall performance. See full result table in A.

is not captured by standard grammaticality benchmark alone.

7 Conclusion

As the BabyLM Challenge extends cognitively-inspired language modeling beyond English, there are methodological challenges in evaluating the formal competence of BabyLM-inspired L2LMs that are modeling second language or bilingual acquisition. To address this, we introduced BLiSS, a large-scale benchmark built on a new paradigm of selective tolerance. By evaluating models on controlled triplets (corrected, learner error, artificial error), BLiSS measures a model's ability to distinguish naturalistic human errors from contrived ones, disentangling sensitivity to learner patterns from general grammatical competence. Our experiments demonstrate that selective tolerance is a distinct capability from standard grammaticality, with performance clustering strongly by training paradigm and revealing sensitivity to L1-specific transfer effects. We hope that BLiSS will serve as both a benchmark and a research catalyst for developing L2 language models that better reflect the diversity and systematicity of human language acquisition.

Limitations

Several limitations should be considered when interpreting our results. First, BLiSS relies on

sentence-level corrections from learner corpora, which may not capture all aspects of learner language development. The benchmark focuses on grammatical and lexical errors but does not assess discourse-level phenomena, pragmatic competence, or other dimensions of L2 proficiency that extend beyond sentence boundaries. This imbalance may affect the reliability of conclusions about advanced learner behavior and limits our ability to study developmental trajectories at higher proficiency levels.

Specific L1 backgrounds and grammatical error types that were already infrequent in the source corpora become even more sparse in the final dataset. The low success rate of our generation process (4.8%) means that only the most common and structurally regular phenomena are represented at scale. This may limit the statistical power for fine-grained analyses on these lower-frequency L1-error combinations and means that our results are most representative of common error patterns.

Acknowledgments

With thanks to Laura Barbenel for proof-reading this manuscript. We thank the anonymous reviewers for their useful feedback and suggestions, which greatly improved the manuscript. This paper reports on work supported by Cambridge University Press & Assessment.

References

Theodora Alexopoulou, Jeroen Geertzen, Anna Korhonen, and Detmar Meurers. 2015. Exploring big educational learner corpora for sla research: Perspectives on relative clauses. *International Journal of Learner Corpus Research*, 1(1):96–129.

Theodora Alexopoulou, Detmar Meurers, and Akira Murakami. 2022. Big data in sla: Advances in methodology and analysis. In *The Routledge handbook of second language acquisition and technology*, pages 92–106. Routledge.

Tatsuya Aoyama and Nathan Schneider. 2024. Modeling Nonnative Sentence Processing with L2 Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4927–4940. Association for Computational Linguistics.

Catherine Arnett, Tyler A. Chang, James A. Michaelov, and Benjamin Bergen. 2025. On the acquisition of shared grammatical representations in bilingual language models. In *Annual Meeting of the Association for Computational Linguistics*.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805. Association for Computational Linguistics.

Harald Clahsen and Claudia Felser. 2006. Continuity and shallow structures in language processing. *Applied Psycholinguistics*, 27(1):107–126.

Ionut Constantinescu, Tiago Pimentel, Ryan Cotterell, and Alex Warstadt. Investigating Critical Period Effects in Language Acquisition through Neural Language Models. 13:96–120.

Stephen Pit Corder. 2015. The significance of learners' errors. In *Error analysis*, pages 19–27. Routledge.

Scott A Crossley and Kristopher Kyle. 2022. 34 managing second language acquisition data with natural. *The Open Handbook of Linguistic Data Management*, page 411.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Linnea Evanson, Yair Lakretz, and Jean-Remi King. 2023. Language acquisition: do children and language models follow similar learning stages? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12205–12218.

Manuel Faysse, Patrick Fernandes, Nuno M. Guerreiro, António Loison, Duarte M. Alves, Caio Corro, Nicolas Boizard, João Alves, Ricardo Rei, Pedro Henrique Martins, Antoni Bigata Casademunt, François Yvon, André Martins, Gautier Viaud, C'eline Hudelot, and Pierre Colombo. 2024. CroissantLLM: A truly bilingual French-English language model. *ArXiv*, abs/2402.00786.

Jeroen Geertzen, Theodora Alexopoulou, and Anna Korhonen. 2014. Automatic linguistic annotation oflarge scale L2 databases: The EF-Cambridge Open Language Database(EFCamDat).

Jaap Jumelet, Abdellah Fourtassi, Akari Haga, Bastian Bunzeck, Bhargav Shandilya, Diana Galvan-Sosa, Faiz Ghifari Haznitrama, Francesca Padovani, Francois Meyer, Hai Hu, Julen Etxaniz, Laurent Prevot, Linyang He, María Grandury, Mila Marcheva, Negar Foroutan, Nikitas Theodoropoulos, Pouya Sadeghi, Siyuan Song, and 7 others. forthcoming. BabyBabelLM: A multilingual benchmark of developmentally plausible training data. *Forthcoming*. Under review. Download PDF available online.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu,

Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Bert Le Bruyn and Magali Paquot. 2021. *Learner corpus research meets second language acquisition*. Cambridge University Press.

Eric H. Lenneberg. 1967. The biological foundations of language. *Hospital Practice*, 2(12):59–67.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The effect of learner corpus size in grammatical error correction of ESL writings. In *Proceedings of COLING 2012: Posters*, pages 863–872, Mumbai, India. The COLING 2012 Organizing Committee.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Diane Nicholls, Andrew Caines, and Paula Buttery. 2024. The Write & Improve Corpus 2024: Error-annotated and CEFR-labelled essays by learners of English.

Miyu Oba, Tatsuki Kuribayashi, Hiroki Ouchi, and Taro Watanabe. 2023. Second language acquisition of neural language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13557–13572, Toronto, Canada. Association for Computational Linguistics.

Adam Pauls and Dan Klein. 2012. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968, Jeju Island, Korea. Association for Computational Linguistics.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is more: Pre-training cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 174–188, Miami, FL, USA. Association for Computational Linguistics.

Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. 2024. Release of pre-trained models for the Japanese language. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13898–13905. https://arxiv.org/abs/2404.01657.

Larry Selinker. 1972. Interlanguage. *International Review of Applied Linguistics in Language Teaching*, 10(1-4):209–232.

Igor Sterner and Simone Teufel. 2025. Minimal pair-based evaluation of code-switching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18575–18598, Vienna, Austria. Association for Computational Linguistics.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Aditya Yadavalli, Alekhya Yadavalli, and Vera Tobin. 2023. SLABERT talk pretty one day: Modeling second language acquisition with BERT. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11763–11777, Toronto, Canada. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Ge Zhang, Scott Qu, Jiaheng Liu, Chenchen Zhang, Chenghua Lin, Chou Leuang Yu, Danny Pan, Esther Cheng, Jie Liu, Qunshu Lin, Raven Yuan, Tuney Zheng, Wei Pang, Xinrun Du, Yiming Liang, Yinghao Ma, Yizhi Li, Ziyang Ma, Bill Lin, and 26 others.

2024. MAP-Neo: Highly capable and transparent bilingual large language model series. *arXiv preprint arXiv: 2405.19327*.

Tianyu Zhao, Toshiaki Wakatsuki, Akio Kaga, Koh Mitsuda, and Kei Sawada. rinna/bilingual-gpt-neox-4b.

A Full Evaluation Results

Model	BLiSS							BLiMP
	HAP		HAP@τ		SO		LP	
	Overall	L1	Overall	L1	Overall	L1		
Bilingual LLMs								
CroissantLLM	67.51*	81.76**	57.42*	71.70**	57.64*	54.09	12.84	81.20
Neox-4B	60.55*	78.26**	42.98*	62.32**	35.15*	1.88	16.27	82.12
MAP-Neo-7B	66.81*	77.12	58.14*	72.03**	56.05*	45.76**	14.14	82.12
Bilingual BabyLM models								
BBLM–DE	60.15*	76.92**	50.59*	66.67**	43.73*	33.33	18.80	66.32
BBLM–ZH	59.56*	72.88**	49.57*	66.95**	34.93*	38.14	20.44	66.44
BBLM–ID	60.08*	66.6	50.41*	62.96	43.66*	37.04	28.57	66.22
BBLM–FR	60.38*	79.25**	50.70*	67.92**	43.93*	44.03	13.71	66.10
SLABERT								
SLABERT–JP	50.42*	47.83	31.40*	27.54	16.58*	15.94	63.46	49.16
SLABERT–FR	52.22*	52.20	34.50*	33.96	16.20*	15.09	57.47	48.44
SLABERT–ID	46.36*	38.89	30.91*	25.93	15.40*	12.96	57.14	51.36
SLABERT–PL	46.01*	54.92**	32.99*	38.52	14.07*	18.85	57.57	52.00
B-GPT								
B-GPT-ES-SIM	66.43*	77.14**	56.48 *	62.14**	54.57*	50.00	12.99	52.66
B-GPT-ES-SEQ	66.06*	74.29**	56.15*	61.43**	55.06	48.57**	12.17	54.19
EFCAMDAT Trained								
LM–EF	51.87*	47.94**	36.94*	33.78**	15.23*	12.66**	39.51	53.76
Noise–EF	47.69*	46.26	28.47*	29.67	11.40*	11.33	41.95	54.84
Contr–EF	62.09*	62.24	48.02*	41.74	23.70*	18.03	69.51	50.04
Compl–EF	49.50*	56.23	44.75*	45.80	21.35*	19.54	40.73	54.74
CLC Trained								
CLC–A1	61.94*	-	51.94*	-	28.86*	-	35.28	54.68
CLC–A2	58.00*	-	46.94*	-	23.74*	-	39.27	54.55
CLC–B1	61.27*	-	50.75*	-	29.62*	-	29.62	53.79
CLC–B2	55.18*	-	43.47*	-	21.93*	-	41.68	54.95
CLC–C1	51.89*	-	38.65*	-	18.48*	-	46.02	52.70

Table 6: BLiSS metrics (HAP, HAP@τ, Strict Order, LP) with L1 and Overall subcolumns, alongside BLiMP grammaticality accuracy. An asterisk (*) indicates performance significantly above the 50% chance baseline ($p < 0.05$), while a double asterisk (**) on L1 scores indicates a statistically significant difference between the L1-specific and overall performance.

B Learner-Trained Model Details

All models were trained using the HuggingFace `Trainer` API with the following configuration. Training ran for 10 epochs for CLC-trained models and 5 epochs for EFCAMDAT-trained models.

Parameter	**Value**
Seed	42
Block size	1024 tokens
Per-device batch size	2
Gradient acc. steps	8
Effective batch size	16
Learning rate	5×10^{-5}
Weight decay	0.1
Warmup steps	500
Logging steps	50
Max steps	-1 (full epochs)
Scheduler	cosine
Optimiser	AdamW
Mixed precision	`fp16`
Gradient checkpointing	Enabled
Save strategy	End of each epoch

Table 7: Training hyperparameters.

C ERRANT Annotation Scheme

All learner sentences in BLiSS are automatically annotated with ERRANT v3.0.0 to obtain token-level error labels.

Table 8: Complete list of valid error code combinations

Operation Tier	Type	Missing	Unnecessary	Replacement
Token Tier	Adjective	M:ADJ	U:ADJ	R:ADJ
	Adverb	M:ADV	U:ADV	R:ADV
	Conjunction	M:CONJ	U:CONJ	R:CONJ
	Determiner	M:DET	U:DET	R:DET
	Noun	M:NOUN	U:NOUN	R:NOUN
	Particle	M:PART	U:PART	R:PART
	Preposition	M:PREP	U:PREP	R:PREP
	Pronoun	M:PRON	U:PRON	R:PRON
	Punctuation	M:PUNCT	U:PUNCT	R:PUNCT
	Verb	M:VERB	U:VERB	R:VERB
	Other	M:CONTR	U:CONTR	R:CONTR
	Morphology	-	-	R:MORPH
	Orthography	-	-	R:ORTH
	Other	M:OTHER	U:OTHER	R:OTHER
	Spelling	-	-	R:SPELL
	Word Order	-	-	R:WO
Morphology Tier	Adjective Form	-	-	R:ADJ:FORM
	Noun Inflection	-	-	R:NOUN:INFL
	Noun Number	-	-	R:NOUN:NUM
	Noun Possessive	M:NOUN:POSS	U:NOUN:POSS	R:NOUN:POSS
	Verb Form	M:VERB:FORM	U:VERB:FORM	R:VERB:FORM
	Verb Inflection	-	-	R:VERB:INFL
	Verb Agreement	-	-	R:VERB:SVA
	Verb Tense	M:VERB:TENSE	U:VERB:TENSE	R:VERB:TENSE

Sample-Efficient Language Modeling with Linear Attention and Lightweight Enhancements

Patrick Haller **Jonas Golde** **Alan Akbik**

Humboldt-Universität zu Berlin
{patrick.haller.1, jonas.max.golde, alan.akbik}@hu-berlin.de

Abstract

We study architectural and optimization techniques for sample-efficient language modeling under the constraints of the BabyLM 2025 shared task. Our model, **BLaLM**, replaces self-attention with a linear-time mLSTM token mixer and explores lightweight enhancements, including short convolutions, sliding window attention with dynamic modulation, and Hedgehog feature maps. To support training in low-resource settings, we curate a high-quality corpus emphasizing readability and pedagogical structure. Experiments across both STRICT and STRICT-SMALL tracks show that (1) linear attention combined with sliding window attention consistently improves zero-shot performance, and (2) the Muon optimizer stabilizes convergence and reduces perplexity over AdamW. These results highlight effective strategies for efficient language modeling without relying on scale.

1 Introduction

Training language models under strict resource constraints remains a central challenge, both for advancing theoretical understanding and for enabling practical deployment on limited hardware. The BabyLM shared task provides a unique opportunity to evaluate models in a controlled setting, where participants are restricted to training on at most 10 million (STRICT-SMALL) or 100 million (STRICT) words for a maximum of 10 epochs. This environment encourages the development of sample-efficient algorithms rather than scale-dependent strategies.

Our submission focuses on algorithmic enhancements rather than introducing novel architectures. Specifically, we examine whether recent advancements in model design and optimization can be adapted to improve sample efficiency when applied to a standard Transformer backbone. Our contributions are as follows:

1. **Model Architecture:** We replace the self-attention mechanism in a standard Transformer with the linear-time mLSTM module, yielding an efficient subquadratic variant we refer to as BLaLM.

2. **Optimization:** We evaluate the *Muon* optimizer, a recently proposed alternative to AdamW, which introduces dynamic momentum and a decoupled weight decay schedule. We compare Muon and AdamW under identical training conditions.

3. **Architectural Enhancements:** We introduce and evaluate several lightweight modifications to the BLaLM model, including sliding window attention (SWA), short convolutional layers, and dynamic attention modulation.

4. **Corpus Construction:** We curate a high-quality corpus by filtering and modifying existing text corpora, aiming to improve training dynamics for small models. Preliminary results indicate improved downstream performance relative to unfiltered datasets.

Our experiments lead to two key findings: First, replacing self-attention with a linear-time mLSTM token mixer, especially when combined with sliding window attention and dynamic modulation, leads to strong zero-shot performance under low-resource constraints. Second, the Muon optimizer improves convergence and stability compared to AdamW, particularly for matrix-shaped parameters. Together, these results point to practical strategies for improving sample efficiency in compact language models.

2 Preliminaries and Related Work

Transformers

The Transformer architecture, proposed by Vaswani et al. (2017), has become the

Proceedings of the First BabyLM Workshop, pages 175–191

de facto standard for large-scale language modeling. Unlike recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997), Transformers process sequential input in parallel through self-attention. Given query, key, and value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d}$, the self-attention output is computed as:

$$\boldsymbol{y} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}} \odot \boldsymbol{M}\right)\boldsymbol{V}, \qquad (1)$$

where $\mathbf{M}$ is a causal mask that prevents attending to future tokens. While highly expressive, self-attention incurs $\mathcal{O}(n^2 d)$ complexity in both computation and memory, which becomes a bottleneck for long sequences, especially during autoregressive decoding.

Linear Attention

To address the quadratic bottleneck, Katharopoulos et al. (2020) proposed linear attention mechanisms that replace the softmax kernel with a feature map $\phi(\cdot)$ such that:

$$\text{softmax}(\mathbf{Q}\mathbf{K}^\top) \approx \phi(\mathbf{Q})\phi(\mathbf{K})^\top. \qquad (2)$$

This formulation enables autoregressive decoding in $\mathcal{O}(nd^2)$ time by exploiting the associativity of matrix multiplication, reducing memory usage and improving scalability.

Linear attention has since been extended in numerous architectures targeting long-context modeling and efficient training (Sun et al., 2023; Poli et al., 2023). In the BabyLM 2024 shared task, Haller et al. (2024) introduced *BabyHGRN*, which leverages a recurrent HGRN2 token mixer within Transformer-style blocks. It achieved competitive results under low-resource constraints, motivating continued exploration of subquadratic alternatives.

xLSTM and mLSTM

xLSTM (Beck et al., 2024) revisits the LSTM architecture with two core innovations: exponential gating and enhanced memory structures. It defines two cells, sLSTM and mLSTM, which are assembled into residual blocks.

mLSTM extends the scalar memory c_t to a matrix memory $\mathbf{C}_t \in \mathbb{R}^{d \times d}$ that stores key-value pairs via an outer-product update. The forget gate f_t acts as a decay, while the input gate i_t controls the learning rate:

$$\mathbf{C}_t = f_t \mathbf{C}_{t-1} + i_t v_t k_t^\top, \quad n_t = f_t n_{t-1} + i_t k_t, \quad (3)$$

and retrieval is computed using:

$$h_t = o_t \odot \frac{\mathbf{C}_t q_t}{\max\{|\langle n_t, q_t \rangle|, 1\}}, \qquad (4)$$

with q_t, k_t, v_t derived from learned projections.

As with other linear-time mechanisms, mLSTM supports parallel training and linear-time autoregressive decoding. It serves as the token mixer in our model architecture.

The BabyLM Benchmark

The BabyLM initiative (Charpentier et al., 2025) introduced a suite of benchmarks for evaluating language models in low-resource conditions, with a focus on learnability, generalization, and alignment with developmental stages. The 2025 shared task continues this focus, imposing strict limits on training data and epochs to emphasize sample efficiency and high quality data curation.

Optimizers

Adaptive optimizers such as Adam (Kingma and Ba, 2017) and its decoupled variant AdamW (Loshchilov and Hutter, 2019) remain standard for LLM training due to their robustness and ease of tuning. However, their dynamics can be suboptimal for matrix-shaped parameters, especially in low-data or large-batch regimes.

Recent alternatives aim to improve convergence and stability, including Lion (Chen et al., 2023), Sophia (Liu et al., 2024), and Shampoo (Gupta et al., 2018). Muon (Keller, 2024) orthogonalizes gradient updates via a truncated Newton-Schulz iteration, improving conditioning for matrix-valued parameters with minimal overhead. It is typically used in hybrid schemes, where scalar parameters (e.g., layer norms, biases) are still optimized with AdamW.

Muon has shown benefits in both vision and language domains (AI et al., 2025; Liu et al., 2025), including better training stability, faster convergence, and improved data efficiency, which all are valuable under the constraints of BabyLM.

3 Data Curation

Data quality plays a critical role in small-scale language modeling, where noisy or incoherent samples can substantially degrade performance. In this work, we prioritize readability, coherence, and syntactic simplicity to improve learnability under low-resource constraints.

Dataset	# Words STRICT-SMALL	# Words STRICT
CHILDES Project (Child-directed speech)	2M	8.7M
Fineweb-Edu	2M	21M
TinyStories	1M	35M
Project Gutenberg, Fiction Books	1.5M	1.7M
Simple Wikipedia (English)	1.5M	22.6M
Cosmopedia		
- WikiHow	1.8M	10.1M
- Math	0.2M	0.3M
Total	$\approx$ 10M	$\approx$ 99.5M

Table 1: Token counts per data source in the curated corpus used for the STRICT-SMALL and STRICT tracks of BabyLM 2025.

Rather than relying solely on large, unfiltered corpora, we curate a dataset by filtering and modifying existing sources using heuristic and LLM-guided approaches. Our filtering pipeline targets syntactically clean, semantically rich, and pedagogically structured documents likely to be learnable by small models.

3.1 Data Sources

Our curated pretraining corpus draws from a diverse set of publicly available datasets selected for their relevance to early language acquisition, general knowledge, and structured instruction. The largest component is **FineWeb-Edu** (Lozhkov et al., 2024; Penedo et al., 2025), a filtered subset of FineWeb-2 annotated for educational value. To incorporate spoken language patterns, we include transcripts from the **CHILDES** corpus (MacWhinney, 2000), which features child-directed speech. We also leverage **TinyStories** (Eldan and Li, 2023), a synthetic story dataset designed for early learners. Fictional content is sourced from a filtered selection of English novels from **Project Gutenberg** (Gerlach and Font-Clos, 2020), while simplified encyclopedic entries come from **Simple Wikipedia**. Finally, we include domain-specific educational content from **Cosmopedia** (Ben Allal et al., 2024), which covers instructional materials such as WikiHow articles and mathematics explanations. A breakdown of word counts per dataset and track is provided in Table 1.

3.2 Filtering Pipeline

We apply dataset-specific filters to improve linguistic quality and reduce noise. Below we summarize our main filtering strategies:

FineWeb-Edu Although FineWeb-Edu is already annotated for educational value, we re-evaluate all samples using our own educational scoring prompt (Appendix A) with LLaMA 3.3–70B.

Gutenberg Fiction We discard Gutenberg entries without named entities and subsample up to 200 samples per book to ensure diversity.

TinyStories We remove template-like introductions (e.g., "Once upon a time…") to reduce repetition and increase stylistic variety.

Simple Wikipedia We retain only paragraphs with at least 15 words to remove boilerplate and fragmented content.

Cosmopedia (WikiHow & Math) We filter for content relevant to K–12 learners and remove excessively long or domain-specific passages.

CHILDES (Child-Directed Speech) CHILDES contains transcripts of parent–child dialogue, marked with speaker tags (e.g., "*MOT:" for mother). We apply:

1. **Speaker Tag Removal:** Prefixes like "*MOT:" or "*COL:" are removed.
2. **Minimum Length Filtering:** We discard utterances with fewer than 7 words.
3. **Grammar Correction:** We normalize speech using LanguageTool for improved grammaticality.

All data is tokenized and counted at the word level to ensure the final corpus respects the BabyLM 2025 limits of 10M (STRICT-SMALL) and 100M (STRICT) words.

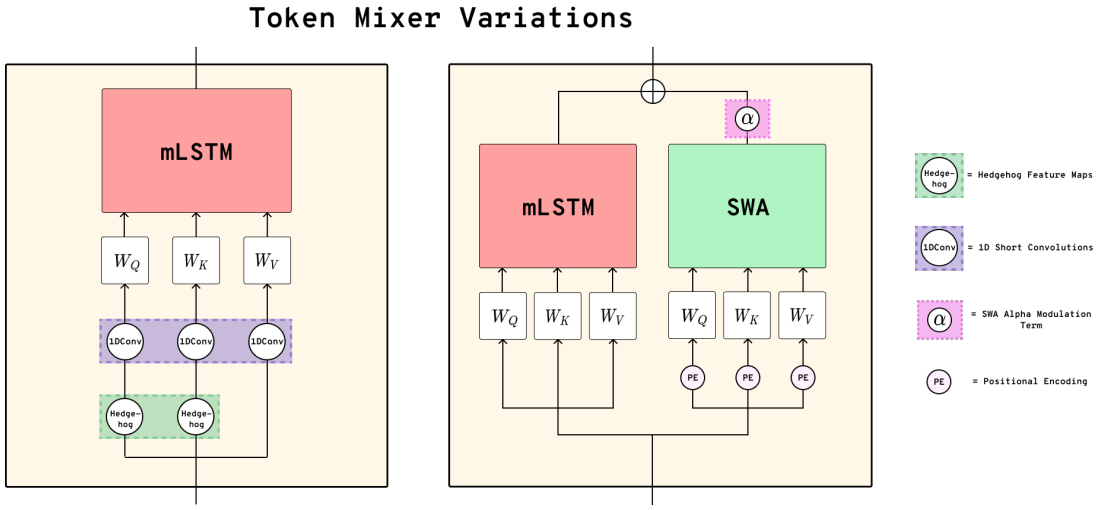


Figure 1: Overview of the BLaLM architecture. The standard self-attention module is replaced by an mLSTM token mixer. Optional enhancements such as sliding window attention (SWA) can be integrated and combined with mLSTM outputs.

4 Model Architecture and Optimization

We aim to evaluate whether architectural and optimization strategies known to improve large-scale language models can also improve sample efficiency under strict training budgets. Rather than designing a novel architecture, we incrementally modify a standard Transformer decoder to assess the contribution of individual components.

Our model, referred to as **BLaLM** (**B**aby **L**inear **A**ttention **LM**), follows the general architecture of recent Qwen models (Bai et al., 2023). It uses **pre-normalization** with **RMSNorm** (Zhang and Sennrich, 2019) for training stability, **feed-forward blocks** with SwiGLU activations, and **rotary positional embeddings (RoPE)** (Su et al., 2023) to encode position information. RoPE is used in both the self-attention baseline and the optional sliding window attention modules in BLaLM.

The key deviation from the standard Transformer lies in the **token mixer**, which is the module responsible for integrating contextual information across tokens. In Transformers, this role is fulfilled by the self-attention mechanism; in BLaLM, we replace it with **mLSTM**, a recurrent linear-time alternative. The mLSTM operates via element-wise gating (forget and input gates) and uses matrix-valued memory updates across learned projections. It supports fully parallel training and linear-time autoregressive decoding, thereby avoiding the quadratic overhead of softmax attention while maintaining expressivity.

This architectural choice preserves full compatibility with Transformer training pipelines, allowing direct comparisons between self-attention and mLSTM-based token mixing.

4.1 Architectural Enhancements

In addition to the mLSTM substitution, we introduce a set of lightweight architectural improvements aimed at enhancing sample efficiency:

- **Short Convolutions (ShortConv):** 1D depthwise convolutions are added before the token mixer on the query and key projections to enhance local inductive bias. Recently added by Gu and Dao (2024); Dao and Gu (2024); Beck et al. (2024); Lan et al. (2025); Nguyen et al. (2025).

- **Sliding Window Attention (SWA)** (Beltagy et al., 2020): A local attention mechanism with fixed-size attention window. Sliding is used in conjunction with the mLSTM token mixer. The input is passed through both modules and added together, like:

$$h_{final} = \frac{h_{LA}}{2} + \frac{h_{SWA}}{2} \tag{5}$$

- **SWA with Dynamic Modulation (DynMod):** Applies a learned gating function to modulate attention hidden states over each layer.

$$h_{total} = h_{LA} + \alpha \cdot h_{SWA} \tag{6}$$

$$h_{total} = h_{LA} + tanh(\alpha) \cdot h_{SWA} \tag{7}$$

Dataset	BLiMP acc.	B. Suppl. acc.	Entity acc.	EWoK acc.	Eye ΔR^2	Reading ΔR^2	Avg.
Baseline-10M	64.96	66.8	40.01	51.55	0.98	0.45	37.45
Baseline-100M	75.68	65.2	34.82	51.82	0.82	0.33	**38.11**
Our-10M	67.99	63.6	39.46	52.27	1.09	0.65	37.51
Our-100M	74.51	57.6	14.65	55.73	1.18	0.90	34.09

Table 2: Comparison of the official BabyLM dataset ("Baseline") and our curated corpus ("Ours") across both strict-small and strict tracks. We report average zero-shot performance; full results in Appendix C.

- **Hedgehog Feature Maps** (Zhang et al., 2024): A recently proposed mechanism that mimics several properties of softmax-based attention. It is applied to the query and key projections.

Each mechanism, as illustrated in Figure 1, is introduced independently and evaluated against the base BLaLM configuration to quantify its contribution under fixed training budgets.

5 Training Setup

All experiments are conducted under the BabyLM 2025 shared task constraints for the STRICT-SMALL (10M words) and STRICT (100M words) tracks, using the curated dataset described in Section 3

Sequence Length and Batching

We train with a context length of 512 tokens and an effective global batch size of 64. When hardware limitations require smaller per-device batches, we use gradient accumulation to match the target batch size. Text data is first concatenated into a continuous stream before splitting into fixed-length sequences to avoid truncation and minimize padding overhead.

Models

We use two architectural variants throughout our experiments: a baseline Transformer decoder (Qwen-style) and our proposed model, BLaLM, which replaces self-attention with an mLSTM token mixer. Both models share the same configuration where applicable; architectural differences are detailed in Appendix B.

Training Duration and Checkpointing

Each model is trained for a maximum of 10 epochs over the respective corpus. We evaluate all saved checkpoints and report results for the best-performing one based on average downstream performance.

Evaluation

All models, except the final submissions, are evaluated using the fast zero-shot evaluation suite provided by the BabyLM organizers (Charpentier et al., 2025).[1] We rely on this fast evaluation method to score all intermediate checkpoints and select the best model per run. Final submissions are evaluated on hidden tasks and additionally fine-tuned on GLUE.[2]

Learning Rate Scheduling

We use a cosine decay schedule with a 10% linear warmup phase. The learning rate used for each experiment is reported in the corresponding results section.

Optimizers

We use either AdamW or Muon, as introduced in Section 2. In Sections 6.1 and 6.2, AdamW is used as the default optimizer. Later experiments switch to Muon, which is applied to matrix-shaped parameters (e.g., projection weights, MLP layers), while AdamW handles all scalar-valued parameters (e.g., embeddings, biases, and normalization layers).

6 Experiments

6.1 Experiment 1: Dataset Performance

This experiment evaluates the impact of our curated dataset relative to the baseline corpus provided by the BabyLM organizers. Since the original training configuration of the baseline models could not

[1] Shortly before the deadline, a bug was discovered in the evaluation for the *WuG* task. Due to time constraints, we were unable to re-evaluate all models. We therefore exclude this task from our reported results.

[2] For a complete list of benchmarks and descriptions, see the official BabyLM 2025 evaluation pipeline.

Track	Model	LR	BLiMP acc.	B. Suppl. acc.	Entity acc.	EWoK acc.	Eye ΔR^2	Reading ΔR^2	Avg.
Strict-Small	Transformer (9)	4e-4	64.95	57.2	18.07	51.36	1.13	0.93	32.27
	BLaLM (9)	5e-4	66.72	55.6	40.93	51.0	0.91	0.60	35.96
Strict	Transformer (9)	4e-4	72.44	62.0	20.63	53.36	1.02	0.74	35.03
	BLaLM (10)	5e-4	74.49	60.4	21.99	53.91	1.03	0.71	35.42

Table 3: Zero-shot performance comparison between Transformer and BLaLM across both BabyLM tracks. Results reflect the best-performing epoch per model in brackets after the model name. See Appendix D for full details.

be fully replicated, particularly in terms of preprocessing, we train our own baseline models using their corpus under our experimental setup for a fair comparison.

Setup We use our proposed architecture (BLaLM) and train two variants on each dataset, the BabyLM-provided corpus and our curated corpus, for both the STRICT-SMALL and STRICT tracks. Each configuration is run twice with identical hyperparameters to control for variance. To keep the comparison controlled, we fix the learning rate at 4×10^{-4} for all runs.

Results Table 2 shows that in the STRICT-SMALL setting, our dataset yields slightly higher average scores (37.51 vs. 37.45), with improvements observed in BLiMP, EWoK, and Entity accuracy. In the STRICT track, the performance gap reverses, the baseline corpus outperforms ours, particularly on BLiMP Supplement and Entity.

These results suggest that dataset quality plays a stronger role in low-resource settings, where clean, coherent input provides better learning signals for small models. While the curated data does not consistently outperform the baseline at larger scales, it performs on par, and slightly better in the strict-small regime, without requiring additional sources or augmentation.

Because this dataset was specifically optimized for educational quality, readability, and structure, we use it for all subsequent experiments.

6.2 Experiment 2: Transformers vs. Linear Attention

This experiment assesses the effect of replacing the standard self-attention mechanism in a Transformer with an mLSTM-based token mixer.

Setup We compare two architectures: a baseline Transformer decoder (following the Qwen configuration) and our proposed model, BLaLM. Both models share the same configuration where applicable, differing only in the token mixer. Due to small differences in parameterization between self-attention and mLSTM, the number of layers is adjusted to keep parameter counts approximately matched. Full architectural details are provided in Appendix B.

Experiments are conducted for both the STRICT-SMALL and STRICT tracks. For each architecture, we train models using three learning rates (3e-4, 4e-4, 5e-4) to account for differences in convergence dynamics.

Results Table 3 presents the evaluation results. In the STRICT-SMALL setting, BLaLM consistently outperforms the Transformer baseline across all learning rates, with the best configuration (5e-4) improving the average score from 32.27 to 35.96.

In the STRICT track, results are more balanced. While the Transformer baseline performs better at some learning rates, BLaLM achieves the highest overall score (35.42 compared to 35.03) showing that the benefits of linear attention persist even in the presence of more data, albeit with smaller margins.

These results support the hypothesis that linear-time alternatives like mLSTM can improve sample efficiency in the low-data regime and remain competitive at larger scales, making them a viable drop-in replacement for self-attention in resource-constrained training scenarios.

6.3 Experiment 3: The Choice of Optimizer

This experiment compares two optimizers for pretraining BLaLM: AdamW, the default choice for Transformer training, and Muon, a recently proposed optimizer designed to improve convergence speed and numerical conditioning for matrix-valued parameters.

Setup AdamW is applied to all parameters, while Muon is used in a hybrid scheme as described

in Section 5. Specifically, Muon updates matrix-shaped parameters such as projections and MLP weights, while scalar-valued parameters (e.g., biases, embeddings, normalization layers) are handled by AdamW.

Experiments are conducted in the STRICT track using a fixed learning rate of 4e-4. Each optimizer is evaluated across three independent runs to account for variability in training and initialization. Performance is measured both in terms of validation perplexity and average zero-shot score.

OPTIMIZER	PPL	AVG.
AdamW	11.21 ±0.11	35.75 ±1.74
Muon	7.95 ±0.15	36.24 ±1.16

Table 4: Validation perplexity and average zero-shot scores across three runs comparing AdamW and Muon optimizers for xLSTM training.

Results Table 4 summarizes the results. Muon achieves a lower average validation perplexity (7.95 ± 0.15) compared to AdamW (11.21 ± 0.11), suggesting more stable and efficient optimization.

Zero-shot performance is slightly higher for Muon (36.24 ± 1.16) than for AdamW (35.75 ± 1.74), although the gap is modest. Notably, Muon exhibits more consistent results across runs, indicating improved training stability.

Overall, these findings suggest that Muon improves convergence and may lead to marginal downstream gains under strict resource constraints. Based on these observations, we use Muon for all subsequent experiments.

6.4 Experiment 4: Learning Rate Sweep

This experiment aims to identify the optimal learning rate for pretraining BLaLM under the BabyLM constraints for both the STRICT-SMALL and STRICT tracks.

Setup We conduct a sweep over learning rates in the range from 2e-4 to 7e-4. Each configuration is trained using the same setup described in Section 5, with Muon as the optimizer and a training budget of 10 epochs.

After the initial sweep, we include one additional intermediate learning rate for each track, selected based on observed trends in the initial results. All models are evaluated based on validation perplexity and average zero-shot score. Full results are provided in Appendix F.

	STRICT-SMALL		STRICT	
LEARNING RATE	PPL.	AVG.	PPL.	AVG.
2e-4	16.41	34.80	9.83	34.28
3e-4	16.41	35.61	8.46	35.82
4e-4	20.01	37.27	8.06	35.08
5e-4	16.41	35.03	7.74	35.82
6e-4	16.61	34.17	7.76	35.06
7e-4	**15.73**	**37.53**	7.64	36.10
Additional Learning Rates				
7.5e-4	14.84	37.04	-	-
5.5e-4	-	-	7.70	**37.49**

Table 5: Results from a learning rate sweep for BLaLM on both tracks. Additional intermediate rates were selected based on observed trends.

Results Table 5 reports evaluation results for all tested learning rates. In the STRICT-SMALL track, the highest average score is achieved at 7e-4 (37.53), while 4e-4 and 5e-4 also perform competitively. A follow-up experiment with 7.5e-4 yields slightly lower performance (37.04), suggesting diminishing returns beyond 7e-4.

In the STRICT track, performance peaks at 5.5e-4 with an average score of 37.49. This outperforms 5e-4 and 7e-4, suggesting 5.5e-4 offers the best trade-off.

Overall, the results highlight that optimal learning rates differ by data scale. In low-resource regimes, higher learning rates such as 7e-4 are beneficial, while in higher-resource settings, more moderate values around 5.5e-4 provide the best trade-off between stability and generalization.

6.5 Experiment 5: Evaluating Lightweight Architectural Enhancements

In this experiment, we augment the base BLaLM architecture with a range of lightweight mechanisms that have shown promise in recent work on efficient sequence modeling. These additions are designed to improve local processing, inductive bias, and compositional mixing.

Setup All experiments are conducted in both the STRICT-SMALL and STRICT tracks using the same training setup as in previous sections. The learning rate is fixed at 4e-4, and the Muon optimizer is used for all runs.

Each enhancement is introduced independently to isolate its effect on performance. In addition, a subset of combinations is also evaluated to test potential synergies between modules. Results are reported in terms of validation perplexity and average zero-shot score across the BabyLM benchmark

Mechanism	strict-small PPL.	strict-small Avg.	strict PPL.	strict Avg.
BLaLM	20.01	**37.27**	7.95	35.08
- *ShortConv*	12.37	36.41	6.48	34.57
- *SWA*	12.08	36.16	7.38	35.86
- *SWA with Memory*	10.08	34.96	6.67	37.21
- *SWA DynMod*	9.44	36.15	7.76	**38.82**
- *SWA DynMod Bounded*	8.58	34.41	6.84	36.21
- *Hedgehog*	6.18	33.58	6.68	36.65
- *Hedgehog + SWA*	7.27	36.25	6.63	34.20

Table 6: Evaluation of lightweight architectural enhancements added to BLaLM. Each mechanism is tested independently on both BabyLM tracks. Results include validation perplexity and average zero-shot performance.

suite.

Results Table 6 presents the results. In the STRICT-SMALL track, most mechanisms improve over the base model, with ShortConv and SWA variants performing particularly well. Hedgehog yields the lowest perplexity (6.18), suggesting improved optimization efficiency, although this does not translate directly into the highest downstream score.

In the STRICT track, the most effective mechanism is SWA combined with dynamic modulation, which reaches the highest average score of 38.82. Hedgehog and bounded DynMod also improve performance relative to the base configuration.

We additionally tracked the learned weights α for SWA in the dynamic modulation setups. As shown in Appendix G, these weights vary across layers and increase over training time, suggesting that deeper layers rely more heavily on local context mixing.

Overall, these results indicate that augmenting mLSTM with lightweight attention or modulation mechanisms can improve both perplexity and downstream performance, particularly when local structure and compositional control are emphasized.

6.6 Final Submission Models

For our final BabyLM 2025 submissions, we select configurations that balance strong downstream performance with stable optimization, as identified in our preceding experiments.

STRICT-SMALL Track (10M words): We use BLaLM with mLSTM token mixing, augmented with short convolutions. The learning rate is set to 7e-4, and optimization uses Muon for matrix-shaped parameters and AdamW for scalars. This configuration yields robust zero-shot accuracy across linguistic and educational benchmarks while maintaining low perplexity.

STRICT Track (100M words): We adopt the same architecture, but with a learning rate of 5.5e-4, which in our sweep showed superior generalization in higher-data regimes. We include SWA with bounded dynamic modulation, avoiding further additions to preserve architectural simplicity.

We denote the models BLaLM-STRICT-SMALL and BLaLM-STRICT respectively.

In both tracks, models are trained for 10 epochs using the curated dataset described in Section 3. Final submissions are fine-tuned on GLUE for hidden test set evaluation, as per shared task protocol.

The results are shown in Table 7.

Model	Zero-Shot Avg.	Fine-Tune Avg.
BLaLM-STRICT-SMALL	29.54	57.35
BLaLM-STRICT	36.49	56.70

Table 7: Final performance of our submitted models (BLaLM-STRICT-SMALL and BLaLM-STRICT) on the full BabyLM and (Super)GLUE benchmark suites. Results are averaged across all tasks.

7 Conclusion

We introduced **BLaLM**, a sample-efficient language model built with linear attention and lightweight enhancements. Across both strict and strict-small tracks, BLaLM outperforms Transformer baselines in low-resource settings and remains competitive at larger scales. Our results highlight two actionable insights: (1) combining mLSTM with sliding window attention and dynamic modulation consistently improves downstream generalization, and (2) the Muon optimizer stabilizes training and reduces perplexity, outperforming AdamW for matrix-valued parameters. These findings offer concrete guidance for efficient model design in data-constrained environments.

References

Essential AI, :, Ishaan Shah, Anthony M. Polloreno, Karl Stratos, Philip Monk, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Ashish Tanwer, Darsh J Shah, Khoi Nguyen, Kurt Smith, Michael Callahan, Michael Pust, Mohit Parmar, Peter Rushton, Platon Mazarakis, Ritvik Kapila, Saurabh Srivastava, Somanshu Singla, Tim Romanski, Yash

Vanjani, and Ashish Vaswani. 2025. Practical efficiency of muon for pretraining.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2024. xlstm: Extended long short-term memory.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. 2023. Symbolic discovery of optimization algorithms.

Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english?

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1).

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces.

Vineet Gupta, Tomer Koren, and Yoram Singer. 2018. Shampoo: Preconditioned stochastic tensor optimization.

Patrick Haller, Jonas Golde, and Alan Akbik. 2024. BabyHGRN: Exploring RNNs for sample-efficient language modeling. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 82–94, Miami, FL, USA. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention.

Jordan Keller. 2024. Muon: A drop-in optimizer for faster convergence. `https://kellerjordan.github.io/posts/muon/`.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Disen Lan, Weigao Sun, Jiaxi Hu, Jusen Du, and Yu Cheng. 2025. Liger: Linearizing large language models to gated recurrent structures.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. 2024. Sophia: A scalable stochastic second-order optimizer for language model pretraining.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. 2025. Muon is scalable for llm training.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu: the finest collection of educational content.

Brian MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk: Transcription format and programs, Vol. 1, 3rd ed.* Lawrence Erlbaum Associates Publishers, Mahwah, NJ, US. This book is the first of two volumes (for volume 2, see record 2000-03631-000) documenting the three components of the CHILDES Project: CHAT transcription manual and CLAN analysis manual. Useful for novice and experienced users, as well as instructors and students studying child language transcripts.

Chien Van Nguyen, Ruiyi Zhang, Hanieh Deilamsalehy, Puneet Mathur, Viet Dac Lai, Haoliang Wang, Jayakumar Subramanian, Ryan A. Rossi, Trung Bui, Nikos Vlassis, Franck Dernoncourt, and Thien Huu Nguyen. 2025. Lizard: An efficient linearization framework for large language models.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von

Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all – adapting pre-training data processing to every language.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. 2024. The hedgehog the porcupine: Expressive linear attentions with softmax mimicry.

A Dataset Curation: Prompt

```
Below is an extract from a web page. Evaluate whether the page has
    ↪ a high educational value and could be useful in an
    ↪ educational setting for teaching from primary school to
    ↪ grade school levels using the additive 5-point scoring
    ↪ system described below. Points are accumulated based on the
    ↪ satisfaction of each criterion:
Scoring Criteria:
- +1 Educational Relevance: The extract contains factual or
    ↪ instructional content related to general knowledge,
    ↪ science, math, language, or other academic domains, even if
    ↪ mixed with irrelevant content like ads or unrelated
    ↪ commentary.

- +1 Coherence and Structure: The extract has a recognizable
    ↪ structure (e.g. paragraphs, bullet points, logical flow)
    ↪ and is written in a mostly coherent and syntactically
    ↪ correct way, even if it includes some tangents or
    ↪ inconsistencies.

- +1 Readability and Simplicity: The language is accessible to
    ↪ grade school students, avoiding technical jargon or overly
    ↪ complex sentence constructions. Sentences are clear,
    ↪ concise, and vocabulary is age-appropriate.

- +1 Explainability and Pedagogical Quality: Concepts are
    ↪ explained, not just stated. The text may include analogies,
    ↪ definitions, or examples that make it easier to understand.
    ↪ It supports comprehension and learning.

- +1 Learnability by Small Models: The extract is particularly
    ↪ suitable for training smaller language models: it avoids
    ↪ long-range dependencies, sticks to one or two topics, and
    ↪ has low noise and high signal. Ideal examples follow a
    ↪ pattern, use repetition to reinforce structure, and do not
    ↪ rely heavily on context outside the extract.

The extract:
{0}

After examining the extract:
- Briefly justify your total score, up to 100 words.
- Conclude with the score using the format: "Educational score:
    ↪ <total points>"
```

Figure 2: LLM-based prompt used to assign a **custom** educational scores to FineWeb-Edu samples. The prompt includes a 5-point additive scoring rubric focusing on pedagogical value, readability, and coherence.

B Model Configurations

Hyperparameter	Value
Hidden Size	1024
Intermediate Size	1536
Num Attention Heads	16
Num Hidden Layers	
- Transformer	26
- BLaLM	24
Vocab Size	15K
Parameter Count	
- Transformer	250M
- BLaLM	270M

Table 8: Model configurations for Transformer and BLaLM. Hidden layer count is adjusted to ensure comparable parameter counts across architectures.

C Experiment 1: Full Results

DATASET	BLIMP acc.	B. SUPPL. acc.	ENTITY acc.	EWOK acc.	EYE ΔR^2	READING ΔR^2	AVG.
Baseline-10M (7)	64.96	66.8	40.01	51.55	0.98	0.45	37.45
Baseline-10M (8)	65.49	60.4	35.22	52.27	0.76	0.4	35.75
Baseline-100M (9)	75.43	63.6	20.47	53.36	0.59	0.26	35.61
Baseline-100M (6)	75.68	65.2	34.82	51.82	0.82	0.33	38.11
Our-10M (8)	67.99	63.6	39.46	52.27	1.09	0.65	37.51
Our-10M (9)	66.81	59.2	41.97	52.73	0.85	0.53	37.01
Our-100M (10)	74.51	57.6	14.65	55.73	1.18	0.9	34.09
Ours-100M (10)	74.6	57.2	16.56	53.64	1.21	0.57	33.96

Table 9: Detailed results comparing our curated dataset to the official BabyLM baseline. The number in parentheses indicates the best-performing training epoch.

D Experiment 2: Full Results

Track	Model	LR	BLiMP acc.	B. Suppl. acc.	Entity acc.	EWoK acc.	Eye ΔR^2	Reading ΔR^2	Avg.
Strict-Small	Transformer (10)	3e-4	64.47	58.8	15.71	49.45	1.06	0.56	31.67
	Transformer (9)	4e-4	64.95	57.2	18.07	51.36	1.13	0.93	32.27
	Transformer (10)	5e-4	65.37	54.4	14.5	52.18	0.83	0.44	31.28
	BLaLM (9)	3e-4	63.07	59.6	38.52	51.27	0.82	0.5	35.63
	BLaLM (9)	4e-4	66.93	55.6	28.74	52.09	0.91	0.5	34.12
	BLaLM (9)	5e-4	66.72	55.6	40.93	51.0	0.91	0.6	35.96
Strict	Transformer (10)	3e-4	72.81	60.8	18.51	55.09	0.82	0.58	34.76
	Transformer (9)	4e-4	72.44	62.0	20.63	53.36	1.02	0.74	35.03
	Transformer (10)	5e-4	72.47	63.2	18.21	51.55	1.11	0.71	34.54
	BLaLM (10)	3e-4	73.40	61.2	14.70	54.55	0.93	0.51	34.21
	BLaLM (10)	4e-4	74.60	57.2	16.56	53.64	1.21	0.57	33.96
	BLaLM (10)	5e-4	74.49	60.4	21.99	53.91	1.03	0.71	35.42

Table 10: Detailed zero-shot results for Transformer and BLaLM across BabyLM benchmarks. Parentheses indicate best-performing epoch.

E Experiment 3: Full Results

Model	BLiMP acc.	B. Suppl. acc.	Entity acc.	EWoK acc.	Eye ΔR^2	Reading ΔR^2	Avg.
AdamW							
Run1 (10)	74.96	60.8	13.25	55.55	0.85	0.61	34.33
Run2 (10)	74.37	65.6	32.8	54.82	0.9	0.68	38.19
Run3 (10)	76.54	61.6	14.72	53.82	0.92	0.57	34.69
Muon							
Run1 (10)	76.4	70.4	22.27	55.91	1.17	0.79	37.82
Run2 (10)	75.82	66.4	15.39	55.73	1.1	0.88	35.88
Run3 (10)	75.27	64.0	15.98	53.18	1.07	0.76	35.03

Table 11: Zero-shot results for xLSTM models trained with AdamW and Muon optimizers (3 runs). Parentheses indicate best-performing epoch.

F Experiment 4: Full Results

Learning Rate	BLiMP acc.	B. Suppl. acc.	Entity acc.	EWoK acc.	Eye ΔR^2	Reading ΔR^2	Avg.
STRICT-SMALL							
1e-4 (6)	57.16	58.4	39.95	52.73	0.28	0.3	36.04
2e-4 (6)	61.37	59.2	41.81	52.55	0.74	0.59	34.80
3e-4 (7)	67.19	60.8	33.44	50.55	1.14	0.58	35.61
4e-4 (8)	69.55	58.0	41.91	52.36	1.17	0.68	37.27
5e-4 (6)	69.12	59.6	26.8	52.91	1.1	0.69	35.03
6e-4 (9)	69.98	61.6	18.89	52.91	1.03	0.65	34.17
7e-4 (7)	70.68	60.4	38.87	53.82	0.96	0.47	37.53
STRICT							
2e-4 (9)	73.81	60.4	17.42	52.64	0.87	0.54	34.28
3e-4 (10)	75.91	67.6	15.06	55.09	0.81	0.5	35.82
4e-4 (8)	66.82	54.8	33.54	54.0	0.92	0.43	35.08
5e-4 (10)	76.25	66.4	15.08	55.55	1.03	0.62	35.82
5.5e-4 (9)	76.1	63.6	28.04	55.64	0.87	0.73	37.49
6e-4 (9)	76.42	59.2	18.64	54.36	0.94	0.83	35.06
7e-4 (9)	75.85	66.0	19.46	53.73	1.03	0.55	36.10

Table 12: Full results for Experiment 4. The number in brackets after each learning rate denotes the best performing epoch.

G Experiment 5: Full Results

Mechanism	BLiMP acc.	B. Suppl. acc.	Entity acc.	EWoK acc.	Eye ΔR^2	Reading ΔR^2	Avg.
strict-small							
ShortConv (6)	67.13	57.6	38.82	52.82	1.42	0.71	36.41
SWA (9)	64.86	54.4	43.17	52.91	1.1	0.52	36.16
SWA With Memory (7)	65.83	52.8	35.97	52.18	1.99	1.04	34.96
SWA DynMod (6)	67.36	54.4	41.28	51.36	1.69	0.83	36.15
SWA DynMod Bounded (7)	65.59	52.4	33.86	52.36	1.35	0.93	34.41
Hedgehog (10)	68.3	53.2	24.5	53.0	1.52	0.99	33.58
Hedgehog + SWA (6)	65.43	54.4	42.49	52.73	1.55	0.94	36.25
strict							
ShortConv (8)	74.31	61.2	15.63	54.18	1.05	1.08	34.57
SWA (8)	74.29	60.8	21.42	56.45	1.2	1.02	35.86
SWA With Memory (10)	71.52	65.2	31.04	54.18	0.85	0.49	37.21
SWA DynMod (9)	76.39	68.0	31.32	55.64	0.83	0.76	38.82
SWA DynMod Bounded (10)	73.64	66.0	22.36	53.55	0.97	0.75	36.21
Hedgehog (8)	74.64	62.0	24.69	56.73	1.32	0.55	36.65
Hedgehog + SWA (7)	72.94	58.8	16.28	54.64	1.69	0.9	34.20

Table 13: Full results for Experiment 5. Parentheses indicate the best-performing epoch per configuration.

G.1 Alpha Value Development for DynMod Runs

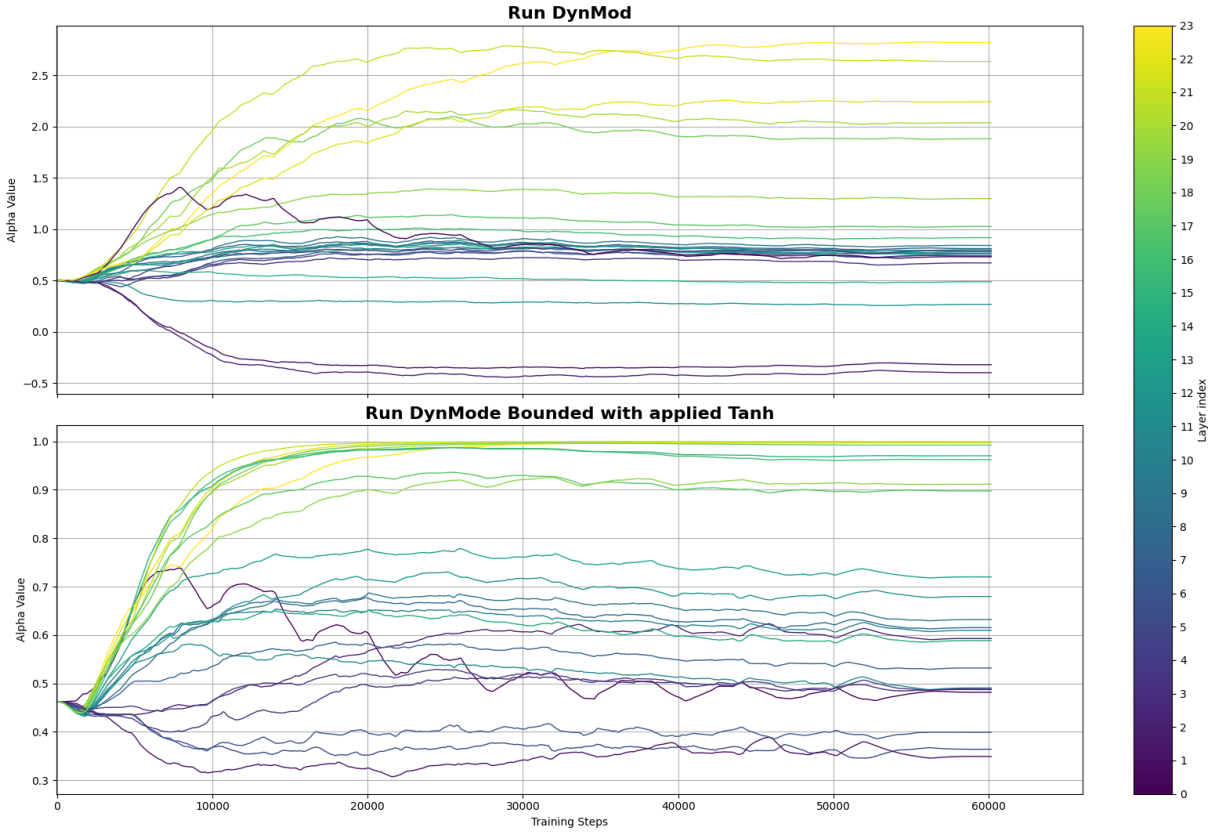


Figure 3: Layer-wise development of dynamic modulation weights () during training for the bounded DynMod variant. We apply the tanh function to stabilize values. Later layers show increased reliance on local mixing.

H Final Submission: Full Results

BENCHMARK	BLaLM STRICT-SMALL	BLaLM STRICT
BLIMP	67.0	74.7
B. SUPPL.	53.3	61.0
ENTITY TRACKING	33.7	22.2
EWOK	50.6	53.6
EYE TRACKING	1.1	1.1
SELF PACED READING	1.0	0.6
WUG ADJ. NORM.	50.3	47.5
WUG. PAST TENSE	-20.7	37.5
COMPS	50.5	58.3
AOA	8.6	8.6
AVERAGE	29.54	36.49

Table 14: Final BabyLM benchmark results for BLaLM-STRICT-SMALLand BLaLM-STRICTmodels. Includes hidden tasks.

MODEL	BOOLQ acc.	MNLI acc.	MRPC acc.	QQP acc.	MULTIRC acc.	RTE acc.	WSC acc.	AVG.
BLaLM-STRICT-SMALL	64.03	34.18	69.60	59.92	57.54	54.67	61.54	57.35
BLaLM-STRICT	64.03	34.27	69.10	58.90	57.54	51.79	61.53	56.70

Table 15: Performance of BLaLM-STRICT-SMALLand BLaLM-STRICTon (Super)GLUE tasks after fine-tuning.

Looking to Learn: Token-wise Dynamic Gating for Low-Resource Vision-Language Modelling

Bianca-Mihaela Ganescu* **Suchir Salhan*** **Andrew Caines** **Paula Buttery**
ALTA Institute Department of Computer Science & Technology, University of Cambridge

Abstract

Training vision-language models on cognitively-plausible amounts of data requires rethinking how models integrate multimodal information. Within the constraints of the Vision track for the BabyLM Challenge 2025, we propose a lightweight decoder-based architecture with (1) token-wise dynamic gating for adaptive fusion of linguistic and visual cues, (2) feature modulation and channel attention to maximise the utility of limited visual information and (3) auxiliary contrastive objectives for visual grounding. Evaluation on five benchmarks (BLiMP, BLiMP Supplement, EWoK, Winoground and VQA) shows competitive or superior performance to multimodal baselines. More notably, our dynamic gate discovers interpretable patterns without explicit supervision, favouring visual cues for content words and linguistic cues for function words. While we identify limitations in the Challenge constraints, such as the information bottleneck created by global image embeddings and training instability from the dataset split, our findings establish dynamic gating as a powerful tool for efficient multimodal learning, offering both interpretability and performance even under severe constraints.

LookingtoLearn on HuggingFace (models, tokenizers, and checkpoints)

Training Code Open-Sourced on GitHub

1 Introduction

Large language models have achieved impressive capabilities, yet their learning process markedly differs from naturalistic human language learning. Children learn their first language from just tens of millions of words (Warstadt et al., 2023; Gilkerson et al., 2017) with minimal supervision, whereas state-of-the-art language models require three to four magnitudes more data (Warstadt et al., 2023).

* **Corresponding Authors:** bmg44@cam.ac.uk, sas245@cam.ac.uk

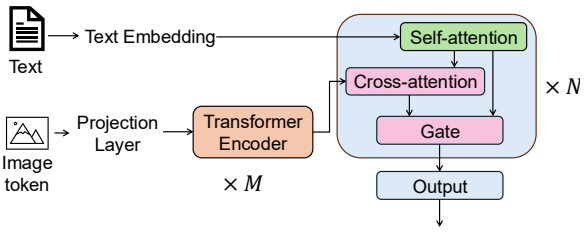


Figure 1: Simplified dual-stream architecture. The **text processing stream** (top) embeds text input tokens and feeds them through an N-layer transformer decoder, which applies masked self-attention, cross-attention to image features and a dynamic gating module to fuse representations. The **image processing stream** (bottom) projects a DINOv2 global token into the same space and processes it with an M-layer transformer encoder. The image processing stream, cross-attention and gating modules are skipped for text-only samples.

Furthermore, human language learning is inherently a multimodal process. Usually, visual experiences play a crucial role in the acquisition of early language and its expansion in the first years of life (Rose et al., 2009; Morgenstern, 2014, 2023; Karadöller et al., 2024). This cognitive reality motivates our work in the BabyLM Challenge Vision track (Charpentier et al., 2025), where we develop a framework inspired by human selective attention that learns *when* and *how* to leverage visual cues during language processing without explicit supervision.

Our proposed solution is a decoder-based vision-language model for which we introduce three key innovations. First, we implement a **dynamic gating mechanism** that learns to selectively weight visual versus linguistic cues for each token based on context. Second, we explore several **feature enhancement techniques** in order to maximise the utility of limited visual information. Third, we investigate the impact of **contrastive learning auxiliary objective functions** that operate at both

Proceedings of the First BabyLM Workshop, pages 192–217

the sentence and word levels under low-resource constraints.

In this work, we aim to answer several key questions:

1. Q: Can dynamic gating mechanisms be repurposed to learn meaningful vision-language fusion patterns without explicit supervision? (Subsection 3.3)
   A: Yes, statistical analysis of our dynamic gate's outputs shows a strong correlation between branch selection and parts-of-speech, as well as a weak correlation between branch selection and concreteness and imageability scores. (Section 5)

2. Q: If so, which linguistic phenomena do our models prioritise visual information for, and does this align with human word grounding?
   A: We find that for parts-of-speech which are open-class and tend to be more grounded (adjective, noun, proper noun, verb) (Haley et al., 2025), the dynamic gate assigns more weight to visual signals than for function words (conjunction, punctuation, symbols, auxiliary verbs, particles) which tend to be less grounded (Haley et al., 2025). (Section 5)

3. Q: Is the setup of the Vision track optimal for multimodal learning? In particular, can architectural mechanisms compensate for the limited visual information provided by global image embeddings? (Subsection 3.4)
   A: In our framework, global image embeddings create an information bottleneck that feature enhancements cannot fully address (Subsection 4.2). Moreover, we find that the split between text-only and image-caption data causes training instability and identify misalignments between the training data and multiple evaluation benchmarks. (Section 6)

4. Q: Do contrastive learning auxiliary objectives help or hinder small vision-language models under significant data constraints? (Subsection 3.5)
   A: Contrastive learning objectives prove counterproductive without sufficient scale for our selected benchmarks. (Subsection 4.3)

Performance analysis on five BabyLM Challenge benchmarks (BLiMP (Warstadt et al., 2020), BLiMP Supplement (Warstadt et al., 2025), EWoK (Ivanova et al., 2024), Winoground (Thrush et al., 2022) and VQA (Goyal et al., 2017)) reveals task-specific benefits of our proposed framework, with our base model achieving competitive or superior performance compared to the multimodal baselines of the 2025 Challenge.

Overall, this work contributes to the broader goal of taking inspiration from human learning for the development of language models, not just in terms of data, but also in their underlying mechanisms. While the results of dynamic gating show that architectural design can lead to meaningful patterns, our findings also reveal which constraints (visual representation, data curriculum, training datasets and evaluation benchmarks) must be addressed next in order to further improve vision-language models based on human learning.

2 Background

2.1 Vision-Language Models

Vision-language models (VLMs) combine an image encoder and (optionally) a text encoder with a multimodal fusion module to learn joint representations for tasks such as captioning, retrieval, and visual question-answering. Although specific VLM architectures vary, most share a **vision encoder** projecting images into embedding features aligned with language model embeddings, often implemented as a Vision Transformer (ViT) patch encoder pre-trained on rich visual datasets (Dosovitskiy et al., 2021), and a **text encoder**. While early VLMs (e.g., CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021)) use both vision and text encoders trained jointly using contrastive learning to align visual and textual representation in a shared latent space (Li et al., 2025), more recent models such as LLaVA (Liu et al., 2023) no longer employ a separate text encoder, but simply use a visual encoder with a text decoder.

2.1.1 GIT and Flamingo Baselines

The baselines used in the Vision track are the Flamingo (Alayrac et al., 2022) and the Generative Image-to-text Transformer (GIT) (Wang et al., 2022) vision-language models.

The GIT (Wang et al., 2022) architecture consists of an image encoder and a text decoder. The image encoder is pre-trained using a contrastive learning objective, and outputs visual features that are linearly projected and concatenated with embedded text tokens to form the input to the decoder.

The entire model is trained using next token prediction, where each token is predicted based on both the preceding text tokens and the visual features.

Flamingo (Alayrac et al., 2022) is a decoder-based multimodal model that interleaves text decoder layers with gated cross-attention dense blocks that incorporate visual input. An image encoder extracts visual features, which a Perceiver Resampler module (Jaegle et al., 2021) compresses into a fixed number of tokens per image. These serve as keys and queries in the gated cross-attention dense layers inserted between language model blocks, where a tahn-gated learnable scalar scales each cross-attention and feed-forward sublayer to control the flow of visual information. The model is trained with next token prediction.

Last year's submissions to the BabyLM Challenge Vision track did not beat the Flamingo and GIT baselines (Hu et al., 2024). AlKhamissi et al. (2024) proposed a self-synthesis strategy for training a BabyLLaMA (Timiryasov and Tastet, 2023) model, comprising four phases ranging from basic language skills to cognitive tasks. Saha et al. (2024) evaluated the effect of curriculum learning on GIT and Flamingo, concluding that benefits were architecture-, training- and task-dependent. Klerings et al. (2024) investigated the role of visual data in language learning for the GIT architecture, finding that visual training data improves model performance on multimodal benchmarks but has no effect on text-only benchmarks. Moreover, they analysed task-specific neuron usage and concluded that their models are highly modular, with visual inputs influencing which components the model uses to process the same text input.

We hypothesise that the GIT and Flamingo models, originally proposed for large-scale training, lack explicit cognitive motivation and may underperform when scaled down, prompting the implementation of our architecture. We point out that our dynamic gating approach contrasts with Flamingo's gated cross-attention dense blocks as follows: firstly, Flamingo applies uniform layer-wise gating parameters across all tokens, whereas our dynamic gate adapts based on individual tokens. Secondly, our model consistently injects visual features at every decoding layer, while Flamingo introduces visual information every few layers, which could limit the model's ability to learn a stable textual representation.

It is worth noting that the 2024 Challenge did not impose a limit on the number of training epochs, and that the 2024 Flamingo and GIT baselines were trained on 20 epochs worth of text-only data and potentially up to 80 epochs worth of image-caption data. Similarly, the 2025 baselines were trained using a 1:4 ratio between text-only and image-caption data, while respecting the 10-epoch training limit.

2.2 Token-Level Integration of Visual and Linguistic Information

In this work, we ask whether dynamically weighting different modalities can be used to improve next token selection in autoregressive models. Wang et al. (2018) and Kiela et al. (2018) asked a similar question about how to dynamically weight linguistic and visual input based on word type. However, their goal was to create static word embeddings relying on weak supervision. In contrast, we propose using dynamic gating as an unsupervised mechanism during autoregressive generation, where the model decides at each step which modality should produce the next token.

It has been shown in cognitive science research that concrete and abstract words are differentiated in human processing (Binder et al., 2005). Wang et al. (2010) found that abstract words primarily activate language-related brain regions, whereas concrete words engage perceptual brain areas. Further work showed that functional Magnetic Resonance Imaging patterns for concrete nouns can be decoded by both linguistic and visual representations (Anderson et al., 2017), but that abstract nouns are only decodable via linguistic representations (Wang et al., 2018).

An advantage has been reported for words judged to be more concrete than those judged to be more abstract in the learning of distributed semantic representations (Bruni et al., 2014; Hill and Korhonen, 2014). This means that language models are likely to perform better on benchmarks containing more concrete words (Pezzelle et al., 2021). As Kiela et al. (2018) observe, there are complex interactions between concreteness and other factors such as word frequency and word class. We examine concreteness and word class separately in this work and leave the study of their interaction, along with word frequency, for future work.

3 Method

3.1 Overview

We present a multimodal framework for the BabyLM Vision track that learns language from

both ungrounded and visually-grounded text data. We assume that the training data and number of training epochs are fixed according to the **constraints of the BabyLM Challenge 2025**, while the architecture and training regime are variables which we aim to optimise. Specifically, we develop a **dual-stream transformer architecture with three key innovations** that aim to mirror human language processing and improve model performance under Challenge constraints:

1. **Cognitive alignment through dynamic gating**: Unlike standard vision-language models that use uniform fusion strategies, we implement a token-wise dynamic gating mechanism with four variants exploring different granularities and decision levels. This mechanism learns to adaptively weight visual versus linguistic information for each token, drawing inspiration from how humans selectively integrate multimodal information.

2. **Maximising limited visual information**: Given the constraint of using only a global image embedding during training, we implement multiple strategies to compensate for limited visual information. These include modulation techniques that dynamically transform features based on cross-modal context, and channel attention to identify salient aspects within the limited visual representation.

3. **Visual grounding via auxiliary objective functions**: we explore two auxiliary objective functions to enhance visual grounding in our framework: (1) a contrastive learning objective (Radford et al., 2021) which aligns entire captions with images at the sentence level, and (2) LexiContrastive Grounding (Zhuang et al., 2024), which performs word-level alignment between individual tokens and images. These auxiliary objectives aim to improve language learning by creating stronger associations between linguistic and visual representations.

3.2 Base Architecture

At the core of our framework, we design an autoregressive dual stream transformer drawing inspiration from the architecture of state-of-the-art vision-language models such as LLaVA (Liu et al., 2023) and QWen-VL (Bai et al., 2023). A simplified illustration of our architecture is shown in Figure 1.

The architecture consists of four main components: (1) a **text processing stream** that uses standard decoder layers with learned embeddings and positional encodings to process both text-only data and image captions; (2) an **image processing stream** that takes DINOv2 embeddings as input, projects them into the model's hidden space and refines them with additional transformer encoder layers for empirical performance, future patch-token compatibility and computational efficiency (details in Appendix I); (3) a **multimodal decoder** integrates text and image features using three sequential mechanisms: masked self-attention applied to the text features, followed by cross-attention fusion between text and image features (when available), and dynamic gating that adaptively determines how much to rely on visual versus linguistic information for each token; (4) an **output projection layer** that maps the decoder outputs back to the vocabulary space for next token prediction.

3.3 Dynamic Gating

While dynamic gating in multimodal AI has primarily focused on classification tasks, demonstrating improved robustness and computational efficiency (Xue and Marculescu, 2023; Wang and Wang, 2024; Xie and Zhang, 2020), we ask whether this idea can be repurposed as a cognitively-motivated mechanism for token selection in multimodal autoregressive models.

Our approach is hypothesis-driven. Dynamic gating has proven effective in other multimodal settings, and large vision-language models appear to exhibit *implicit* gating capabilities through attention patterns learned at scale. We therefore ask: does introducing an *explicit* token-wise gating mechanism help smaller, data-efficient models achieve similar selective integration more reliably?

Our hypothesis is therefore as follows: just as human language processing selectively integrates visual information, for example, relying heavily on visual inputs for concrete, perceptual words (e.g., “dog", “red") while defaulting to linguistic knowledge for abstract terms (e.g., “therefore", “impossible"), a token-wise dynamic gating mechanism could teach a model to make similar fine-grained fusion decisions. By conditioning each gate on both the current text hidden state and the cross-attention features, the model can learn to amplify the vision input when it truly informs the next word and ignore it when it does not.

We implement four variants of a dynamic gating

mechanism, varying along two axes: **(1) granularity**, whether the gate is computed per feature or per token, and **(2) soft vs hard**, whether the gate outputs continuous weights or discrete decisions. The granularity axis investigates whether different tokens require different subsets of visual features (e.g., colour features for "red", spatial features for "above") or whether coarse per-token gating is sufficient. The soft vs hard gating axis examines whether binary selection or continuous weighting of features yields more interpretable fusion patterns and better performance.

More specifically, the four gating variants compute gating weights g to dynamically fuse text and cross-attention representations via

$$h_{\text{fused}} = g \odot h_{\text{text}} + \big(1 - g\big) \odot h_{\text{crossAttn}} \quad (1)$$

and differ in whether g operates per feature or per token and makes continuous or discrete selections. The technical implementation details for the four gating variants are available in Appendix A.

3.4 Feature Representation

The BabyLM Challenge provides the images in the training data as single global embeddings. While computationally efficient, this approach limits the spatial visual information available to the model. Traditional vision-language models benefit from patch token representations that preserve spatial information and enable fine-grained visual grounding (Dosovitskiy et al., 2021). Therefore, the next aspects we investigate in this work are methods of maximising the utility of the global image tokens provided in the Challenge.

We explore two complementary modulation techniques, FiLM (Perez et al., 2018) and DyIntra (Gao et al., 2019), which dynamically reshape one set of features based on another, as well as a global channel-attention enhancement. These approaches target different aspects of the representation bottleneck: modulation techniques address cross-modal feature interaction, while channel attention addresses intra-modal feature refinement.

While the dynamic gating mechanism determines *how much* information to incorporate from each modality, FiLM and DyIntra determine *how* that information should be transformed, and a channel attention mechanism determines *what* is meaningful within the image features.

We evaluate these methods at several integration points within our architecture to determine which approach most effectively compensates for the lack of spatial visual information.

The technical details of our implementations are available in Appendix B.

3.5 Auxiliary Objective Functions

As previous work in vision-language models suggests (Lu et al., 2020), a multi-task objective can improve model performance. In this work, we explore training our models using two auxiliary functions, Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) and LexiContrastive Grounding (LCG) (Zhuang et al., 2024). Both functions aim to ground textual representations in visual concepts through contrastive learning, creating a shared embedding space where semantically related image-text pairs are positioned closer together. However, they operate at different levels of granularity: CLIP aligns entire captions with their corresponding images at the sentence level, while LCG performs alignment at the word level between individual tokens and images.

Recent research shows that visual grounding at both sentence and word levels can improve word acquisition in low-data regimes (Zhuang et al., 2023). A CLIP objective could capture global associations that support contextual understanding, while an LCG objective might reflect fine-grained grounded learning. However, as we discuss in the results section (Section 4.3), the effectiveness of these auxiliary objectives significantly depends on various factors, including the visual representation format, batch size and training data.

The technical details of each auxiliary objective function are available in Section C.

4 Results

To evaluate our framework, we select five of the benchmarks proposed in the BabyLM Challenge: BLiMP (Warstadt et al., 2020) and BLiMP Supplement for grammar, EWoK (Ivanova et al., 2024) for world knowledge, Winoground (Thrush et al., 2022) for vision-linguistic compositional reasoning and VQA (Goyal et al., 2017) for image-based question answering. For the first four we used the 2025 evaluation pipeline[1], while for VQA we used the 2024 repository[2]. A detailed description of each benchmark is available in Appendix D.

[1] https://github.com/babylm/evaluation-pipeline-2025

[2] https://github.com/babylm/evaluation-pipeline-2024

Model	BLiMP	BLiMP Supplement	EWoK	Winoground	VQA*
Baselines 2025					
Flamingo (BabyLM Challenge 2025)	70.9	65.1	51	54.8	43.31
GIT (BabyLM Challenge 2025)	72.2	66.4	51.8	56.2	49.82
Baselines 2024					
Flamingo (BabyLM Challenge 2024)	70.9	65.0	52.7	51.6	52.3
GIT (BabyLM Challenge 2024)	65.2	62.7	52.4	55.5	54.1
BabyLLaMA (AlKhamissi et al., 2024)	72.9	54.2	50.2	50.9	42.0
$\text{Flamingo}_{\text{CL}}$ T+C (Saha et al., 2024)	60.13	53.28	50.71	50.80	40.85
GIT_{CL} T+C (Saha et al., 2024)	64.05	51.24	50.98	55.23	43.98
GIT 1/0.25 (Klerings et al., 2024)	71.2	64.6	52.5	56.2	52.2
GIT 1/0.125 (Klerings et al., 2024)	66.3	61.7	52.3	57.0	52.6
Our framework					
Base, soft gate per feature	74.33	56.36	50.81	51.61	50.02
Architectural features					
Soft gate per token	73.86	55.43	51.56	52.14	48.39
Hard gate per feature	74.10	54.16	51.20	50.13	45.62
Hard gate per token	74.19	54.59	51.16	50.80	45.51
No gate	74.70	55.75	50.77	51.34	50.58
FiLM on text	74.32	55.10	50.61	53.49	46.04
FiLM on cross-attention	74.95	56.36	51.62	52.68	49.66
FiLM on image	73.80	54.59	51.06	50.13	17.92
DyIntra on text	74	56.97	51.73	51.47	47.16
DyIntra on cross-attention	73.68	56.68	51.57	53.22	48.87
DyIntra on image	74.69	56.57	51.28	50.00	45.61
Channel attention	74.24	54.23	51.15	51.15	49.15
Auxiliary objective functions					
NTP + CLIP	72.28	54.35	51.45	51.47	47.72
NTP + LCG	70.27	56.91	49.74	50.00	36.62

Table 1: Performance of our base model and variants on five BabyLM Challenge benchmarks. Scores for our models and 2025 baselines are computed using the 2025 evaluation pipeline (BLiMP, BLiMP-S, EWoK, Winoground) and 2024 pipeline (VQA). Green shading indicates performance above the 2025 baselines.

For all architectural features and training strategies we define, we conduct experiments in the form of ablation studies in order to evaluate each potential improvement in isolation. A summary of all the experiments we define is available in Table 6. We train all our models in the same conditions (as described in Table 5) using the same model hyperparameters (summarised in Table 4), with the exception of the auxiliary objective function, for which we increase the batch size from 64 to 128 as a larger batch size is recommended for contrastive learning (Chen et al., 2020).

Our key observations from the complete set of results (Table 1) are summarised below.

4.1 Baselines

Our framework achieves a higher score on BLiMP (almost 4% higher than Flamingo and over 2% higher than GIT) and competitive scores for EWoK, Winoground and VQA. We suggest that our base model outperforms Flamingo and GIT on BLiMP due to architectural differences. These include the clear separation between the text and image streams and consistent fusion in our base model. In our proposed architecture, the first decoder layer's self-attention module processes only textual input, whereas GIT concatenates the image token(s) with the text input before they are fed into the model, which could introduce noise when extracting linguistic signals. Our model consistently integrates visual features at each decoding layer, whereas Flamingo incorporates visual information only at intermittent layers, which could affect the model's ability to learn a robust textual representation.

The lower performance of our model on BLiMP Supplement is due to differences in training data. As shown in Appendix F, the image–caption dataset supports this benchmark far better than the text-only dataset. This favours Flamingo and GIT, trained with a 1:4 text-only/image–caption ratio versus our 1:1 ratio. Winoground shows a similar trend, with baselines benefiting from more image–caption training epochs.

4.2 Performance of Architectural Features

The dynamic gating modules maintain the performance of our base model without gating on

BLiMP and BLiMP Supplement, bring modest benefits for Winoground and show mixed results on VQA. As shown in Table 1, our dynamic gating modules do not have a significant effect on the model's performance on BLiMP and BLiMP Supplement, which is the desired outcome for the text-only benchmarks. There is very little variation in the EWoK scores across the models, which we attribute to a mismatch between training and evaluation data and further discuss in Section 7.

The *soft gate* and *hard gate per token* models outperform the *no gate* model on Winoground. We hypothesise that the gating mechanism in these models produces slightly cleaner, more discriminative joint representations between images and text, which in turn yields a small but consistent improvement.

We observe limited performance benefits of the gating modules on VQA, with the *hard gate* models achieving a lower score ($\sim 5\%$ lower) compared to the *no gate* variant. We hypothesise that the *hard gates* may have learned to allow for stronger image signals than is optimal for VQA, especially since the image-captioning training set contains few constructions similar to VQA (see Section 7).

Modulation and channel attention achieve mixed results over the five benchmarks, underscoring that the global image embedding represents a performance bottleneck. Across the seven variants we implement, no single feature representation technique uniformly improves all five benchmarks (Table 1). FiLM applied to textual representation and cross-attention, along with DyIntra applied to cross-attention, shows modest improvements on Winoground (+1.88%, +1.07%, and +1.61% respectively) by potentially creating more separable joint representations.

With varied impact, all techniques decrease the performance of our base model on VQA. Nevertheless, the cross-attention modulation and our attention seem to preserve most of the linguistic and visual signals needed for VQA, while also bringing slight improvements on Winoground. The results collectively demonstrate that feature modulation and channel attention techniques designed for rich representations show limited and task-specific benefits when applied to severely compressed representations. While certain combinations can enhance performance on specific benchmarks, they cannot overcome the information bottleneck caused by using only a global image embedding.

4.3 Performance of Auxiliary Objectives

A pure next token prediction objective function achieves the best scores for our base model overall. There are multiple potential causes for these results. First, the BLiMP benchmarks rely solely on linguistic information, therefore any auxiliary objective that competes with next token prediction can dilute the model's focus on linguistic signals. This is reflected in the BLiMP score differences of 2.05% and 4.06% with the CLIP and LCG auxiliary functions, respectively.

Second, the CLIP objective was designed for a larger batch size than we could use with our computational budget and more data than the available samples in the image-caption dataset, which may have led to a limited impact. Third, the global image embeddings provide limited visual information, which seems to be insufficient to enable the contrastive auxiliary objectives to make fine-grained visual-linguistic alignments.

Fourth, the alternation between text-only and image-caption epochs may cause training instability, since the auxiliary functions are only used during the image-caption epochs. Therefore, with a 10-epoch budget and the limited global image embeddings, there is no evident benefit of using contrastive learning auxiliary objectives for the benchmarks we selected.

From a cognitive perspective, these negative results may align better with theories of human language acquisition. Children do not learn language through explicit contrastive mechanisms where they simultaneously process what words do and do not mean across hundreds of examples, but rather in rich, multimodal contexts where meaning emerges from use rather than from explicit positive or negative examples. These results support our focus on architectural innovations, such as dynamic gating, which better capture the adaptive nature of human cognitive processing during language learning.

5 Interpretability

Figure 2 illustrates our model's gate value for next token prediction, aggregated per part-of-speech (PoS). The model is evaluated on held-out sentences from the Localized Narratives dataset, amounting to 1,034 tokens. The parts-of-speech for each sentence were extracted using the spaCy English tagger (`en_core_web_sm`) (Honnibal et al., 2020). The gate value plotted is the mean over the gate values per feature. A lower score means the

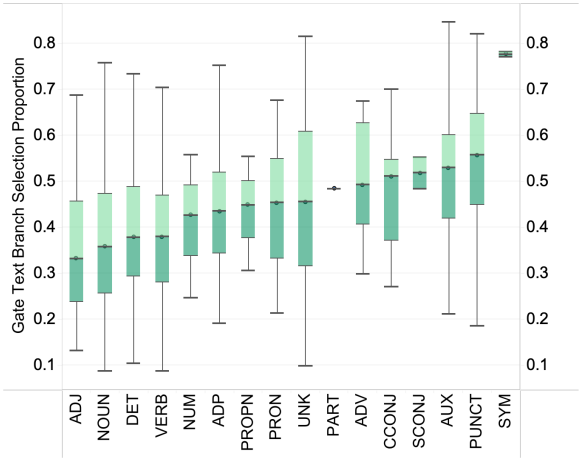


Figure 2: Our base model's aggregated gate values per part-of-speech for next token prediction, based on the Localized Narratives dataset. The parts-of-speech for each sentence were extracted using the spaCy English tagger (Honnibal et al., 2020). Lower values on the y-axis mean that the model attended less to the pure linguistic signals and more to the fused image-text representation when predicting the next token.

model attended less to the pure linguistic signals and more to the fused image-text representation when predicting the next token.

There is an interpretable correlation between gate selection and part-of-speech. For the parts-of-speech which are open-class and generally more grounded (adjective, noun, proper noun, verb) (Haley et al., 2025), the model attends more to the image signals (left side of the plot), while for function words (conjunction, punctuation, symbols, auxiliary verbs, particles) the model attends more to the pure text. Furthermore, the model shows increased visual grounding for numerals, determiners and adpositions, suggesting they leverage visual information for counting and quantity ("two", "three"), uniqueness ("a" vs "the"), spatial reference ("this" vs "that") and spatial relationships ("on", "in", "around"). We confirm the correlation between gate selection and parts-of-speech by running the Kruskal-Wallis statistical test (McKight and Najab, 2010), and obtaining $(H = 154.91, p < 0.001)$.

We also find a statistically significant negative correlation between gate values and concreteness ($\rho = -0.139, p < 0.001$) and imageability ($\rho = -0.153, p < 0.001$) scores using the MRC Psycholinguistic Database (Coltheart, 1981) and Spearman's rank correlation test.

In Table 2, we illustrate the correlation between gate selection and concreteness by aggregating the gate values and defining meaningful categories based on score distributions from the MRC database using cutpoints at mean ± 1 SD. As shown, the correlation is weak ($|\rho| < 0.2$), and the pattern is non-monotonic i.e., moderately abstract/concrete words show higher gate rates than the very abstract/concrete ones, suggesting that other factors, such as part-of-speech, are more important in gating decisions. Similar results for imageability are available in Appendix G.

Category	**Gate Selection**	
	Mean (SD)	# words
Very Abstract (<318)	0.427 (0.141)	420
Abstract (318-438)	0.471 (0.136)	82
Concrete (438-558)	0.391 (0.155)	80
Very Concrete (>558)	0.343 (0.139)	119 9

Categories defined as $\mu \pm \sigma$ based on the MRC database. SD = standard deviation.

Table 2: Mean gate value per concreteness bin for our base model (incorporating a soft gate per feature).

6 Discussion

Missing modality problem. In our experiments, we find that the split of the training data into text-only and image-caption datasets introduces complexity and instability during training. While we attempt to mitigate this by alternating epochs, the approach still yields performance oscillations (details in Appendices F and E). The BabyLM Challenge baselines address this problem by pairing text-only and image-caption in the same batch. However, this results in training the models on four times more image-caption samples than text-only samples. Moreover, to the best of our knowledge, there is no cognitive justification for this split.

Data Curriculum. We explore multiple data curriculum strategies that could optimise learning in our proposed framework. At a coarse-grained level, we explore different orderings between text-only and image-caption epochs. At a fine-grained level, we explore how mixing text-only and image-caption data within the same batch, either uniformly or non-uniformly, impacts the training dynamics and generalisation of the model. Empirical results suggest that alternating between image-caption and text-only epochs is the best strategy for our framework for the five benchmarks we selected in this work. A comprehensive discussion and results are available in Appendix E.

Future work. Based on the results and findings in this work, for the BabyLM Challenge Vision track, we make several observations for future

work. First, the training dataset should be varied, with high-quality text that covers a range of English constructions. In particular, the training dataset should cover constructions (e.g., images paired with question-answers for VQA) and concepts (e.g., EWoK) present in the evaluation benchmarks. Second, for training stability and improved language acquisition, it may be more beneficial to train the model on a completely multimodal dataset, which is one promising avenue for future work. Third, given the limitations that the global image token introduced in this work, future work should use patch-token representations for the image input in order to enable richer multimodal learning – that which is the aim of future iterations of this framework. Finally, it would be interesting to develop benchmarks that specifically reward cognitively-plausible mechanisms: i.e., evaluating the cognitive principles guiding the model's responses.

7 Conclusion

In this work, we show that token-wise dynamic gating enables small vision-language models to adaptively integrate linguistic and visual cues, yielding interpretable patterns and competitive performance under the BabyLM Challenge Vision track constraints. Our results highlight the promise of cognitively-inspired architectural design, while underscoring the need to address limitations in visual representation, training data and evaluation benchmarks to realise the full potential of multimodal learning in low-resource settings.

Limitations

Due to computational constraints, we could not use a larger batch size, which would have benefited the models trained with a contrastive loss objective (Chen et al., 2020).

Two of the BabyLM Challenge benchmarks we used in this work showed limitations in evaluating our multimodal models, which potentially stem from a mismatch with the training data.

Throughout our experiments, we find that EWoK demonstrates no sensitivity to changes in architecture or training strategy, with performance remaining around 50% regardless of the experiment conditions. We therefore investigate the frequency of concepts tested by EWoK in the BabyLM Challenge training data, as previous research suggests that language models rely on frequency more than children do in word acquisition (Chang and Bergen, 2022). A Regular Expression match for the concepts tested in EWoK over the training data revealed that in 37.69% of the EWoK examples, at least one out of two concepts tested appears fewer than 100 times in the training data, with 13% of test examples having both concepts appearing 0 times. Therefore, we conclude that the training dataset does not properly support EWoK evaluation.

By alternating between image-caption and text-only epoch, we also find score differences between epoch types for VQA (details in Appendix F). Our results suggest that VQA depends significantly on the presence of question-answer and turn-taking formats in the training dataset. This finding aligns with observations by Laurençon et al. (2024), who note that vision-language models typically only learn visual question answering during fine-tuning stages, not during pre-training, unless they are explicitly exposed to data following the VQA format. This is particularly problematic under the constraints of the BabyLM Challenge, where no fine-tuning stage exists, forcing models to acquire question-answering capabilities just from pre-training data that lacks examples similar to the VQA task.

As a general observation, training vision-language BabyLMs differs from training state-of-the-art large vision-language models, which rely on large pre-trained components. Moreover, VLMs can undergo multiple training stages where components are selectively frozen or unfrozen, higher-quality data is gradually introduced and the image resolution is progressively increased (Laurençon et al., 2024). With limited data and a maximum of just 10 training epochs under the BabyLM Challenge constraints, implementing multi-stage training strategies becomes significantly more difficult.

Acknowledgments

This paper reports on work supported by Cambridge University Press & Assessment. It was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council. We also particularly thank Dr Diana Galvan-Sosa.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Badr AlKhamissi, Yingtian Tang, Abdülkadir Gökce, Johannes Mehrer, and Martin Schrimpf. 2024. Dreaming Out Loud: A Self-Synthesis Approach For Training Vision-Language Models With Developmentally Plausible Data. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 244–251, Miami, FL, USA. Association for Computational Linguistics.

Andrew J Anderson, Douwe Kiela, Stephen Clark, and Massimo Poesio. 2017. Visually grounded and textual semantic models differentially decode brain activity associated with concrete and abstract nouns. *Transactions of the Association for Computational Linguistics*, 5:17–30.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *Preprint*, arXiv:2308.12966.

Jeffrey R Binder, Chris F Westbury, Kristen A McKiernan, Edward T Possing, and David A Medler. 2005. Distinct brain systems for processing concrete and abstract concepts. *Journal of Cognitive Neuroscience*, 17:905–917.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Tyler A Chang and Benjamin K Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, and 1 others. 2025. BabyLM Turns 3: Call for papers for the 2025 BabyLM workshop. *arXiv preprint arXiv:2502.10645*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Leshem Choshen, Ryan Cotterell, Michael Y Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [call for papers] The 2nd BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2404.06214*.

Max Coltheart. 1981. The MRC psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR)*.

Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. 2019. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6639–6648.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized Project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Jill Gilkerson, Jeffrey A Richards, Steven F Warren, Judith K Montgomery, Charles R Greenwood, D Kimbrough Oller, John HL Hansen, and Terrance D Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American journal of speech-language pathology*, 26(2):248–265.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Coleman Haley, Sharon Goldwater, and Edoardo Ponti. 2025. A grounded typology of word classes. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10380–10399, Albuquerque, New Mexico. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Felix Hill and Anna Korhonen. 2014. Learning Abstract Concept Embeddings from Multi-Modal Data: Since You Probably Can't See What I Mean. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 255–265, Doha, Qatar. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, and 1 others. 2020. spaCy: Industrial-strength natural language processing in python.

Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox, editors. 2024. *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Miami, FL, USA.

Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. 2021. PyTorch. *Programming with TensorFlow: solution for edge computing applications*, pages 87–104.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, and 1 others. 2024. Elements of World Knowledge (EWOK): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Dilay Z Karadöller, Beyza Sümer, and Aslı Özyürek. 2024. First-language acquisition in a multimodal language framework: Insights from speech, gesture, and sign. *First Language*, page 01427237241290678.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic Meta-Embeddings for Improved Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1477, Brussels, Belgium. Association for Computational Linguistics.

Alina Klerings, Christian Bartelt, and Aaron Mueller. 2024. Developmentally Plausible Multimodal Language Models Are Highly Modular. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 118–139, Miami, FL, USA. Association for Computational Linguistics.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, and 1 others. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981.

Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. 2024. Building and better understanding vision-language models: insights and future directions. In *Workshop on Responsibly Building the Next Generation of Multimodal Foundational Models*.

Zongxia Li, Xiyang Wu, Hongyang Du, Huy Nghiem, and Guangyao Shi. 2025. Benchmark evaluations, applications, and challenges of large vision language models: A survey. *arXiv preprint arXiv:2501.02189*, 1.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10437–10446.

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Patrick E McKight and Julius Najab. 2010. Kruskal-wallis test. *The corsini encyclopedia of psychology*, pages 1–1.

Aliyah Morgenstern. 2014. Children's multimodal language development. *Manual of language acquisition*, 2:123–142.

Aliyah Morgenstern. 2023. Children's multimodal language development from an interactional, usage-based, and cognitive perspective. *Wiley Interdisciplinary Reviews: Cognitive Science*, 14(2):e1631.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, and 1 others. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Sandro Pezzelle, Ece Takmaz, and Raquel Fernández. 2021. Word Representation Learning in Multimodal Pre-Trained Transformers: An Intrinsic Evaluation. *Transactions of the Association for Computational Linguistics*, 9:1563–1579.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Susan A Rose, Judith F Feldman, and Jeffery J Jankowski. 2009. A cognitive approach to the development of early language. *Child development*, 80(1):134–150.

Rohan Saha, Abrar Fahim, Alona Fyshe, and Alex Murphy. 2024. Exploring Curriculum Learning for Vision-Language Tasks: A Study on Small-Scale Multimodal Training. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 65–81, Miami, FL, USA. Association for Computational Linguistics.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. 2022. B2t connection: Serving stability and performance in deep transformers. *arXiv preprint arXiv:2206.00330*.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248.

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. *arXiv preprint arXiv:2308.02019*.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

Jing Wang, Julie A Conder, David N Blitzer, and Svetlana V Shinkareva. 2010. Neural representation of abstract and concrete concepts: A meta-analysis of neuroimaging studies. *Human brain mapping*, 31(10):1459–1468.

Nan Wang and Qi Wang. 2024. Dynamic Weighted Gating for Enhanced Cross-Modal Interaction in Multimodal Sentiment Analysis. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(1):1–19.

Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2018. Learning multimodal word representation via dynamic fusion methods. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023. Call for Papers – the BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. arXiv preprint arXiv:2301.11796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjape, Adina Williams, Tal Linzen, and 1 others. 2025. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. *arXiv preprint arXiv:2504.08165*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Long-Fei Xie and Xu-Yao Zhang. 2020. Gate-fusion transformer for multimodal sentiment analysis. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 28–40. Springer.

Zihui Xue and Radu Marculescu. 2023. Dynamic multimodal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2575–2584.

Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. 2023. Visual grounding helps learn word meanings in low-data regimes. *arXiv preprint arXiv:2310.13257*.

Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. 2024. Lexicon-Level Contrastive Visual-Grounding Improves Language Modeling. *arXiv preprint arXiv:2403.14551*.

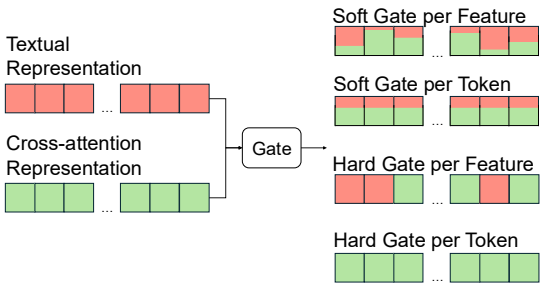


Figure 3: Conceptual output of different gating strategies for fusing textual and cross-attention representations. Each rectangular box represents a token, with the cells within representing dimensions. Red represents textual features, green represents cross-attention features and mixed colours represent fused features. Soft gates apply continuous weights, while hard gates make binary decisions, either per-feature (each dimension independently) or per-token (all dimensions together).

A Dynamic Gating

We define four dynamic gating variants that operate at different granularity and decision levels: *soft gate per feature*, *soft gate per token*, *hard gate per feature*, *hard gate per token*.

Input and Output. All four versions of the dynamic gate have the same input and output. Let $h_{\text{text}} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ be the text hidden states after self-attention and $h_{\text{crossAttn}} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ be the output of the cross-attention between text and image, where B is the batch size and T is the sequence length. Then, the input to the dynamic gate is the concatenation of two hidden representations, $[h_{\text{text}}; h_{\text{crossAttn}}] \in \mathbb{R}^{B\times T\times 2d_{\text{model}}}$. The output is represented by $h_{\text{fused}} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$, which combined the pure linguistic representation with the visually-enriched representation based on the gating weights. In the case of a text-only input to the model, the dynamic gate module is skipped, and h_{text} flows directly through the residual connection.

Figure 3 illustrates the conceptual output for each type of gate.

A.1 Soft Gate per Feature

This variant computes a continuous weight for each feature dimension $i \in \{0, ..., d_{\text{model}} - 1\}$ using the sigmoid function. Concretely, the gate vector is computed as:

$$g = \sigma\big(\text{Linear}\,[\,h_{\text{text}};\, h_{\text{crossAttn}}\,]\big) \in [0,1]^{B\times T\times d_{\text{model}}} \tag{2}$$

The dynamically fused representation is then calculated as:

$$h_{\text{fused}} = g \odot h_{\text{text}} + (1 - g) \odot h_{\text{crossAttn}} \quad (3)$$

We use this variant of dynamic gating in the base model.

A.2 Soft Gate per Token

The soft gate per token calculates a single continuous weight (a scalar), which we apply to all features in the hidden representations:

$$g = \sigma\Big(\text{Linear}\big([\, h_{\text{text}}; h_{\text{crossAttn}}]\big)\Big) \in [0,1]^{B\times T\times 1},$$
$$h_{\text{fused}} = g \odot h_{\text{text}} + \big(1 - g\big) \odot h_{\text{crossAttn}} \quad (4)$$

A.3 Hard Gate per Feature

Drawing inspiration from Xue and Marculescu (2023), we extend the soft gating variants to a hard selection mechanism using the Gumble-Softmax reparametrisation trick (Jang et al., 2016).

The hard gate per feature variant enforces each dimension to choose completely between linguistic or visually-enriched representations. We first compute a 2-way discrete choice for the two hidden representations using a linear layer:

$$g = \text{Linear}\big([\, h_{\text{text}}; h_{\text{crossAttn}}\,]\big) \in [0,1]^{B\times T\times d_{\text{model}}\times 2} \quad (5)$$

Each pair of logits $(l_{b,t,i,0}, l_{b,t,i,1})$ corresponds to the scores for "use h_{text}" versus "use $h_{\text{crossAttn}}$" for feature i at position (b,t). A straightforward hard gate would then be:

$$g_{b,t,i} = \arg\max\big(l_{b,t,i,0},\, l_{b,t,i,1}\big) \quad (6)$$

However, since g is a one-hot vector, it is not differentiable. Therefore we employ a soft gate $\tilde{g}$ during training using Gumble-Softmax, similar to Xue and Marculescu (2023), to enable back-propagation:

$$\tilde{l}_{b,t,i,j} = \frac{l_{b,t,i,j} + z_{b,t,i,j}}{\tau},\; z_{b,t,i,j} \sim \text{Gumbel}(0,1) \quad (7)$$

where τ is the Softmax temperature. We then apply Softmax over the two classes and select the probability corresponding to h_{text} as the soft gate:

$$y_{b,t,i,j} = \frac{\exp\big(\tilde{l}_{b,t,i,j}\big)}{\sum_{k=0}^{1}\exp\big(\tilde{l}_{b,t,i,k}\big)},\; y \in [0,1]^{B\times T\times d_{\text{model}}\times 2} \quad (8)$$

$$\tilde{g}_{b,t,i} = y_{b,t,i,0},\; \tilde{g} \in [0,1]^{B\times T\times d_{\text{model}}} \quad (9)$$

h_{fused} is then be computed as:

$$h_{\text{fused}} = \tilde{g} \odot h_{\text{text}} + \big(1 - \tilde{g}\big) \odot h_{\text{crossAttn}} \quad (10)$$

During training, we anneal the Softmax temperature τ from 1.0 to 0.1 over 80% of the training steps of an image-caption epoch, gradually transitioning from soft to nearly discrete selection. During inference, we convert $\tilde{g}$ to a true one-hot gate g using $\arg\max$.

A.4 Hard Gate per Token

In the per token variant of the hard gate, we collapse the feature-wise gate into a single binary decision. The calculations, training and inference remain the same as in subsection A.3, yet the shape of the parameters changes. The summary of the calculations in this variant is as follows:

$$l = \text{Linear}\big([\, h_{\text{text}}; h_{\text{crossAttn}}]\big) \in \mathbb{R}^{B\times T\times 2} \quad (11)$$

$$y = \text{GumbelSoftmax}\big(l, \tau\big) \in \mathbb{R}^{B\times T\times 2} \quad (12)$$

$$\tilde{g}_{b,t} = y_{b,t,0} \in [0,1] \quad (13)$$

$$h_{\text{fused}} = \tilde{g} \odot h_{\text{text}} + \big(1 - \tilde{g}\big) \odot h_{\text{crossAttn}} \quad (14)$$

where the scalar $\tilde{g}$ is broadcasted over all d_{model} features.

B Feature Representation

B.1 Feature-wise Linear Modulation (FiLM)

To address the limited representational capacity of a single image token, we incorporate Feature-wise Linear Modulation (FiLM) (Perez et al., 2018) as an intra-modal conditioning mechanism. FiLM modulates neural network features through a feature-wise affine transformation, enabling one modality or context to dynamically influence another. Specifically, it applies scaling and shifting to a feature map based on a conditioning input, and can be easily implemented in transformers as follows:

Let $h_{m_1}, h_{m_2} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ be hidden state feature representations with m_1 indicating the primary features and m_2 the conditioning features, $m_1 \neq m_2$. Then,

$$\text{FiLM}\big(h_{m_1}, h_{m_2}\big) = \gamma \odot h_{m_1} + \beta \quad (15)$$

where $\gamma, \beta \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ are scaling and shifting parameters predicted by linear layers from h_{m_2}.

B.2 Dynamic Intra Modulation (DyIntra)

Alternatively to FiLM, we explore the DyIntra module proposed in (Gao et al., 2019), a scaling mechanism that modulates primary features using conditioning features via a simple gating mask. DyIntra predicts a positive-only gain for each representation, allowing it to boost its own hidden features based on cross-modal context without shifting.

Formally, let $h_{m_1}, h_{m_2} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ be hidden state feature representations, $m_1 \neq m_2$. Then, DyIntra computes:

$$m = \sigma\big(\text{Linear}(h_{m_2})\big) \in [0,1]^{B\times T\times d_{\text{model}}} \quad (16)$$

$$\text{DyIntra}\big(h_{m_1}, h_{m_2}\big) = (1+m) \odot h_{m_1} \quad (17)$$

Choosing m_1 and m_2. There are several points in our base model where we could integrate a FiLM or DyIntra modulation module. We evaluate and motivate three such choices as follows:

1. m_1 = self-attention (output), m_2 = image: modulating the text self-attention output with visual features may allow the model to adjust how text tokens relate to each other based on visual context;

2. m_1=cross-attention (output), m_2 = image: modulating cross-attention features with the original image may refine the vision-language fusion by emphasising features that align with the global visual representation;

3. m_1 = image, m_2 = text: modulating image features based on textual context may allow the model to dynamically highlight relevant visual information for the current linguistic processing needs.

B.3 Channel Attention

To implement channel attention for only one image token, we use the Excitation formula from the Squeeze-and-Excitation method (Hu et al., 2018) as follows:

$$h'_{\text{image}} = \sigma\Big(W_2\,\text{ReLU}\big(W_1\,h_{\text{image}}\big)\Big) \odot h_{\text{image}},$$
$$W_1 \in \mathbb{R}^{(d_{\text{model}}/r\times d_{\text{model}})},\ W_2 \in \mathbb{R}^{(d_{\text{model}}\times d_{\text{model}}/r)} \quad (18)$$

where h_{image} is the output of the image encoder and $r = 16$ is the reduction ratio. We expect this method to help the model focus on the most informative features of the visual embedding, improving the quality of image representations.

C Auxiliary Objective Functions

C.1 Contrastive Language-Image Pre-training (CLIP)

We incorporate the Contrastive Language-Image Pre-training (CLIP) objective function into the training of our base model for image-caption epochs steps, as follows:

For each sample in a batch, we first extract pooled representations from both image and text modalities. For text,we compute mean pooling over the the output of the text embedding module, which we denote t_{pooled}. For the image, we use the output of the image encoder directly as its length is 1, i_{pooled}. Both t_{pooled} and i_{pooled} are then projected to a shared contrastive embedding space through specific linear projection layers. We L2-normalise both representations before computing similarity scores. The contrastive loss is formulated as a bidirectional InfoNCE objective (Oord et al., 2018) with learnable temperature τ. It combines text-to-image and image-to-text matching losses, where each direction maximises the similarity between matched pairs while minimising similarity with all other pairs in the batch. The final loss is computed as $\mathcal{L}_{\text{contrastive}} = \frac{1}{2}(\mathcal{L}_{\text{t2i}} + \mathcal{L}_{\text{i2t}})$.

The complete training objective then becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NTP}} + \lambda\mathcal{L}_{\text{contrastive}} \quad (19)$$

where λ represents the weight of the contrastive loss and NTP stands for *next-token prediction*.

In our experiments, we initialise τ to 0.07 and constraint it between 0.05 and 1 during training for stability, and set λ to 1.

C.2 LexiContrastive Grounding (LCG)

LexiContrastive Grounding (LCG) (Zhuang et al., 2024) is a training procedure that implements a grounded language learning objective similar to CLIP. While CLIP operates at sentence level, LCG computes similarity scores at the word level. To calculate the cross-modality contrastive learning loss, LCG leverages the first hidden layer of a model, which stores lexical information. The authors also limit the attention mask applied to the first layer to a previous two-word window in order to encode less linguistic context. The contrastive loss is then calculated per batch from all the token-level representations outputted by the first layer.

For our model, we adapt and implement the LCG during the image-caption epochs as follows:

Let $(\text{text}_i, \text{image}_i)$ represent the image-caption pairs in a batch, where $i \in \{1, 2, \ldots, n\}$ and n is the batch size. Each caption text_i contains m_i tokens: $(t_{i,1}, t_{i,2}, \ldots, t_{i,m_i})$.

To obtain lexically-focused representations, we extract the textual representation from the first layer after the residual connection applied to self-attention:

$$h_1(\text{text}_i) = \text{text}_i + \text{SelfAttn}(\text{LayerNorm}(\text{text}_i)) \tag{20}$$

In our implementation, we experimented with applying a narrow two-word attention mask, however, we noticed conflicts with the next token prediction loss. Specifically, applying the two-word attention mask in the first layer was preventing the next token prediction loss from decreasing. We tried applying the two-word attention mask solely to extract the hidden representation, then switching to the original causal mask for the rest of first layer's forward pass, as well as skipping the cross-modal fusion in the first layer, but neither approach fixed the problem. Therefore, we decided to use the standard causal attention mask when extracting the first layer textual hidden representation, as ablation studies in the original research (Zhuang et al., 2024) did not indicate a significant loss in performance for this case.

Let $h_1(\text{text}_i, j) \in \mathbb{R}^{d_{\text{model}}}$ be the first layer representation of the j-th token (the j-th row of the matrix) in the i-th caption and $\text{enc}(\text{image}_i) \in \mathbb{R}^{d_{\text{model}}}$ represent the output of the image encoder for the i-th image. Then, the matching score between the j-th token in the caption k and image i is calculated as:

$$s(i,j,k) = \frac{(M_{\text{image}} \cdot \text{enc}(\text{image}_i))^T \cdot (M_{\text{text}} \cdot h_1(\text{text}_k, j))}{\tau} \tag{21}$$

where $M_{\text{image}}, M_{\text{text}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are learned projection matrices and τ is a learnable temperature parameter, which we clamp between $[0.05, 2.0]$ for training stability.

For each valid token position, we then compute the LCG contrastive learning loss as:

$$\mathcal{L}_{\text{LCG}} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \mathbf{1}_{\text{valid}}(i,j) \cdot \frac{1}{2}\left[\ell_1(i,j) + \ell_2(i,j)\right] \tag{22}$$

where $\mathbf{1}_{\text{valid}}(i,j)$ is an indicator function for non-padded tokens, and:

$$\ell_1(i,j) = \frac{e^{s(i,j,i)}}{\sum_{k=1}^{n} e^{s(k,j,i)}}, \quad \ell_2(i,j) = \frac{e^{s(i,j,i)}}{\text{neg}(i,j)} \tag{23}$$

The negative term $\text{neg}(i,j)$ is defined as:

$$\text{neg}(i,j) = e^{s(i,j,i)} + \sum_{\substack{k=1 \\ k \neq i}}^{n} \sum_{o=1}^{m_k} \mathbf{1}_{\text{valid}}(k,o) \cdot e^{s(i,o,k)} \tag{24}$$

The total loss combines next-token prediction with word-level contrastive learning loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NTP}} + \lambda \cdot \mathcal{L}_{\text{LCG}} \tag{25}$$

where λ is a hyperparameter controlling the strength of visual grounding. We set λ to 1 through trial and error such that $\mathcal{L}_{\text{NTP}}$ and $\mathcal{L}_{\text{LCG}}$ have the same magnitude.

We use auxiliary functions only during the image-caption epochs, as the image processing stream is skipped for text-only samples.

D Evaluation Benchmarks and Training Data

The evaluation pipeline of the BabyLM Challenge consists of both text-only and multimodal benchmarks. To evaluate our models, we use the following benchmarks from the BabyLM Challenge:

- BLiMP (The Benchmark of Linguistic Minimal Pairs) (Warstadt et al., 2020) evaluates the linguistic abilities of language models through grammatical acceptability judgements. It consists of minimal pairs of sentences testing a specific phenomenon in syntax, semantics or morphology. Each pair contains one well-formed sentence and one ungrammatical sentence. Models are evaluated by checking whether they assign a higher probability to the grammatical sentence in each pair.
- BLiMP Supplement is a held-out evaluation set introduced in the BabyLM Challenge, consisting of five additional linguistic tasks.
- Elements of World Knowledge (EWoK) (Ivanova et al., 2024) is a zero-shot benchmark that targets specific world concepts such as social interactions, spatial relations and physical dynamics. It uses minimal pairs of context-target combinations, where the same target sentence is plausible given one context but

implausible given another. Models are evaluated by checking whether they assign a higher probability to the correct context-target pair.

- Winoground (Thrush et al., 2022) evaluates visio-linguistic compositional reasoning in vision-language models. The dataset consists of hand-curated examples where models must correctly match two images with two captions that contain identical words but in different orders (e.g., “some plants surrounding a lightbulb" vs “a lightbulb surrounding some plants"). Models are evaluated by checking whether they assign a higher probability to the correct caption given the input image.

- VQA v2.0 (Goyal et al., 2017) is an evaluation dataset containing pairs of similar images with identical questions but different correct answers, which forces models to ground their responses in visual content rather than rely on linguistic priors alone. Questions cover multiple categories, such as object recognition, counting and spatial reasoning. Models are evaluated based on which answer they assign the highest probability given the input image and question.

The BabyLM Challenge organisers provide an image-text pre-training dataset for the Vision track, which we use in our work. This dataset consists of two parts: text-only data and text-image data, each containing approximately 50 million words.

The text-only dataset is a subset of the training data proposed for the BabyLM Challenge text-only track. The organisers argue that this dataset is cognitively plausible, consisting of child-directed speech (CHILDES (MacWhinney, 2000)), dialogue (British National Corpus (BNC) conversation section[3], Switchboard Dialog Act Corpus (Stolcke et al., 2000)), children’s stories (Project Gutenberg (Gerlach and Font-Clos, 2020)), movie subtitles (OpenSubtitles (Lison and Tiedemann, 2016)) and Wikipedia[4] content.

The multimodal dataset consists of image-caption pairs selected from the Conceptual Captions 3M dataset (Sharma et al., 2018), and the MS-COCO (Lin et al., 2014) and Open Images (Kuznetsova et al., 2020) subsets of the Localized Narratives dataset (Pont-Tuset et al., 2020). The Conceptual Captions dataset consists of millions of images paired with natural language descriptions automatically scraped, cleaned and filtered from web image alt-text, while the Localized Narratives dataset contains image-caption pairs manually annotated with synchronised mouse traces that spatially ground each word or phrase to specific regions in the image. The images are provided in both raw format and as visual embeddings computed by a visual model using DINOv2 (Choshen et al., 2024; Oquab et al., 2023), a state-of-the-art unsupervised learning algorithm. We use these visual embeddings in both our training and evaluation due to computational constraints.

[3] http://www.natcorp.ox.ac.uk

[4] https://www.wikipedia.org

E Data Curriculum

Since the training dataset consists of both text-only data and image-caption data, each accounting for 50M words, we implement and analyse multiple coarse-grained and fine-grained data curriculum strategies for training.

Coarse-grained epochs: We load the text-only and image-caption data in separate PyTorch (Imambi et al., 2021) data loaders, where each data loader alone is used for one epoch. For the 10 epochs constraint of the BabyLM Challenge, this results in 10 text-only epochs and 10 image-caption epochs. We then experiment with the following:

1. Alternating between image-caption epochs and text-only epochs;

2. Training on all text-only epochs first, then on the image-caption epochs;

3. Training on all image-caption epochs first, then on the text-only epochs.

Fine-grained epochs: For the fine-grained epochs, we define the following two training strategies:

1. We load both the text-only data and the image-caption data in the same data loader, where we pair the text-only data with image tensors filled with 0s for uniformity. The cross-modality path is still skipped in the text-only samples. The original text data is provided in *.txt* files, and we process each text line as one sample. For the image-caption data, we process each (image, caption) pair as one sample. In this setting, the text-only data has twice as many samples as the image-caption data. Therefore, loading and shuffling them in the

same data loader results in a non-uniform distribution between the two and more unstable training.

2. For a uniform distribution between the image-caption data and the text-only data, we take inspiration from the GitHub repository[5] used to train the BabyLM 2024 Challenge baselines, where the authors pair each text-only input with one image-caption input in the same batch sample, resulting in uniform batches. Therefore, in one training step, we perform two forward passes: one using the text-only input and one using the image-caption input. We then sum the losses from each pass and do one backward propagation using the total loss. However, since there are twice as many text-only samples than image-caption samples, this results in training the model twice on the image-caption dataset. For 10 training epochs, this equals 10 text-only epochs and 20 image-caption epochs.

For a fair comparison among all of the methods we implement in this work, we alternate between text-only epochs and image-caption epochs in our experiments exploring architectural changes and auxiliary objective functions. That is because the contrastive learning objective functions compute similarity scores between a caption and all the images in a batch. If the batch contains (many) text-only samples, it cancels the effect of the auxiliary losses.

Empirical results we obtain support this choice among the data curriculum strategies we define. Figure 4 visualises the scores of our base model for the different data curriculum strategies evaluated using the BabyLM Challenge 2024 evaluation pipeline[6].

For BLiMP, the pattern in subfigure 4a suggests that (1) the text-only dataset supports the BLiMP benchmark far more than the image-caption one, and (2) these strategies result in catastrophic forgetting for the model by the end of training.

For BLiMP Supplement, the optimal data curriculum strategy is less clear, as the model's performance oscillates when alternating between epoch types. Comparing the *text-only epochs first* and *image-caption epochs first* strategies shows that the image-caption dataset better supports the model on BLiMP Supplement than the text-only dataset. Interestingly, the model's performance score consistently decreases over checkpoints when the model is trained using the *non-uniform mixed* strategy. A possible explanation for this result is that since there are more text-only samples in a batch than image-caption samples, the gradient updates are dominated by the text-only data, reducing the effect of the image-caption samples.

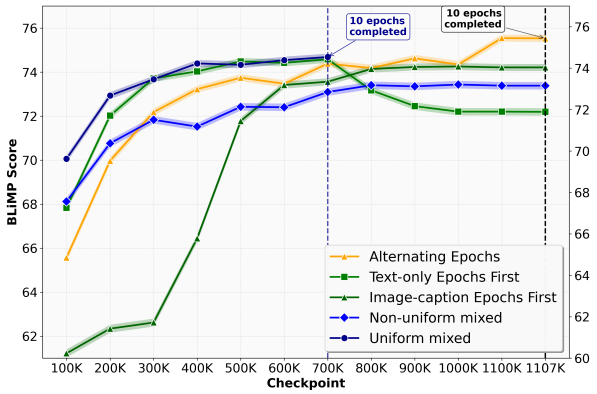
Similar to the other analyses in this work, changing the data curriculum strategy has no visible effect on the EWoK benchmark, underscoring that the training data might not be well-suited for this benchmark.

Training our base model using the *uniform mixed* strategy results in a higher Winoground score, with several checkpoints achieving over 53% on this benchmark. However, a significant factor contributing to this result is the amount of image-caption training data, which is double for this strategy than for the others. Comparing the *text-only epochs first* and *image-caption epochs first* strategies, it can be seen that the model performs better on Winoground when consistently trained on the image-caption dataset. Using the *non-uniform mixed* strategy results in a more unstable performance and a lower final score, possibly due to the dominance of text-only samples in the training batches.

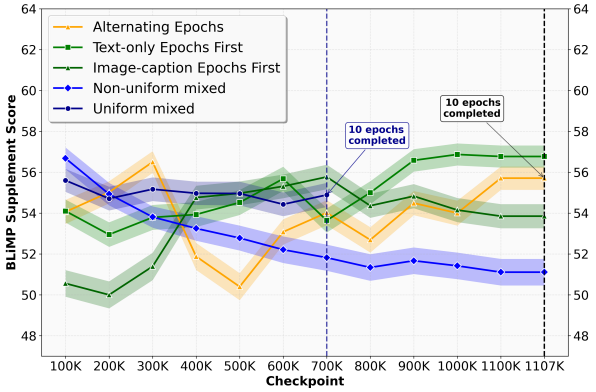
As shown in subfigure 4e, the alternating between text-only and image-caption epochs strategy achieves the best performance on VQA. There is a significant performance gap between the model trained using *alternating epochs* compared to the *mixed* strategies (over 5%), as well as the *text-only epochs first* and *image-caption epochs first* strategies (over 10%). The *alternating epochs* strategy shows an almost consistent increase over checkpoints, whereas the model's performance in the *mixed* variants remains flat, and decreases for the *text-only epochs first* and *image-caption epochs first* strategies. The results of the coarse-grained strategies are likely due to the training data. The image-caption dataset supports the visual reasoning component of VQA, while the text-only dataset supports the question format by containing turn-taking constructions and a significantly larger number of questions than the other dataset. Training the model consistently on only one epoch type deprives it of one of these complementary components. Training by alternating between epoch types appears to strike a balance and avoid catastrophic

[5] https://github.com/aaronmueller/babylm_multimodal_training

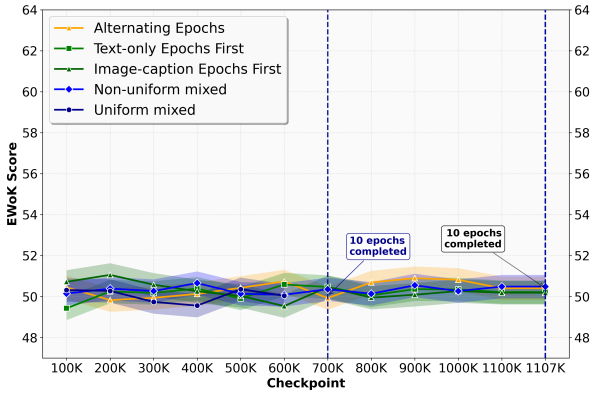
[6] https://github.com/babylm/evaluation-pipeline-2024



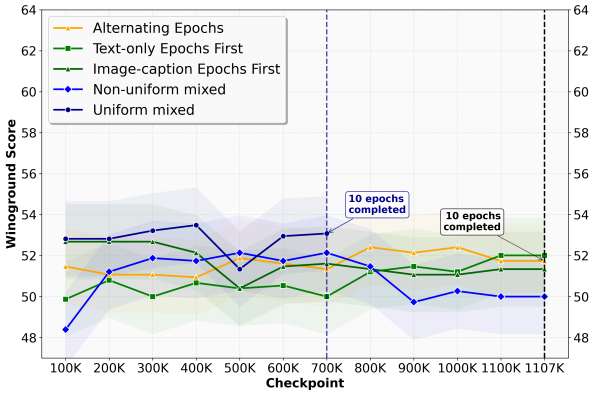
(a) BLiMP scores.



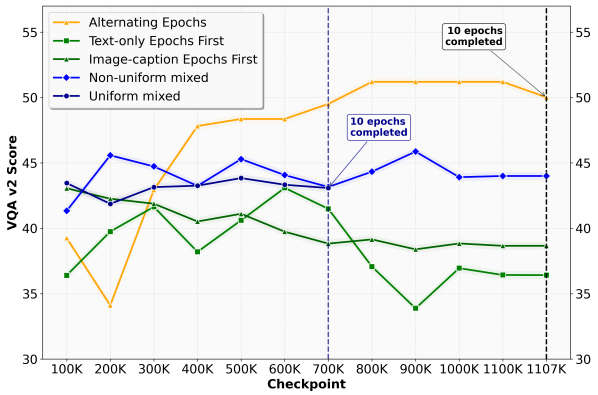
(b) BLiMP Supplement scores.



(c) EWoK scores.



(d) Winoground scores.



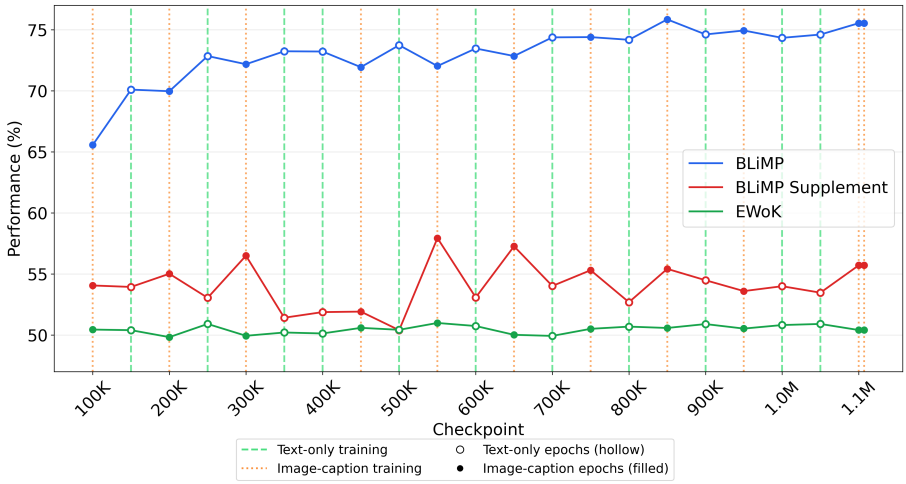
(e) VQA scores.

Figure 4: The performance of our base model on BLiMP, BLiMP Supplement, EWoK, Winoground and VQA for different data curriculum strategies, evaluated using the BabyLM Challenge 2024 evaluation pipeline. The *uniform mixed* strategy follows a different definition than the others, where the number of steps in an epoch equals the number of text-only samples. This results in ~700K training steps for 10 epochs, which are marked in the graphs by the blue dashed line. The end of 10 epochs for the other data curriculum strategies is marked by the black dashed line at step ~1107K.

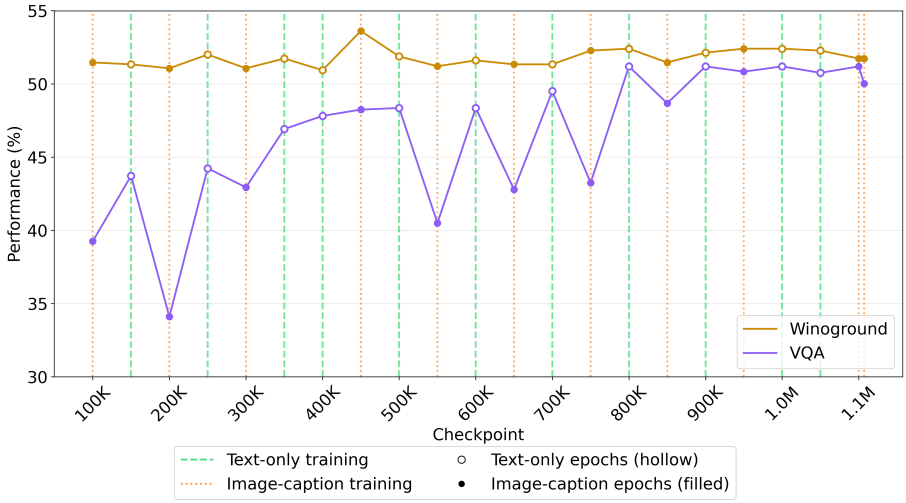
forgetting. The results of the *uniform mixed* strategy are slightly surprising given that the Flamingo and GIT baselines achieve higher VQA scores using this approach, however, the difference could stem from using a lower learning rate and a different text-only to image-caption data ratio.

F Training Dynamics

In figure 5, we visualise the performance of our base model, evaluated every 50,000 steps on the five benchmarks using the BabyLM Challenge 2024 evaluation pipeline, when alternating between text-only and image-caption epochs. The brown dotted lines indicate that the checkpoint was saved



(a) BLiMP, BLiMP Supplement and EWoK scores.



(b) Winoground and VQA scores.

Figure 5: The performance of our base model every 50,000 steps on the BLiMP, BLiMP Supplement, EWoK, Winoground and VQA benchmarks. The brown doted lines indicate that the checkpoint was saved during a text-only epoch, while the green dashed lines indicate that the checkpoint was saved during an image-caption epoch.

during a text-only epoch, while the green dashed lines indicate that the checkpoint was saved during an image-caption epoch. For BLIMP Supplement and VQA, an interesting pattern emerges: the performance scores significantly oscillate based on the type of data on which the model was last trained.

The data that the model was last trained on can be regarded as a fine-tuning step. Thus, we make the following observations:

The base model achieves better performance on BLiMP Supplement during image-caption epochs. As shown in subfigure 5a, our base model obtains higher scores on BLiMP Supplement, a text-only benchmark evaluating grammar, at checkpoints saved during image-caption epochs compared to text-only epochs. We investigate the breakdown of the BLiMP Supplement scores and notice that the score difference for different epoch types stems from two subtasks, *subject-auxiliary inversion* and *turn-taking*. For these subtasks, the performance of our base model fluctuates by even ~10% between checkpoints. We thus investigate the log probability scores of our base model for each subtask example at checkpoints 500,000 (text-only epoch) and 550,000 (image-caption epoch), for which the former model is incorrect and the latter is correct. These two checkpoints present the highest difference in BLiMP Supplement scores (7.54%). We make the following observations:

1. For the *subject-auxiliary inversion*, 68.2% of the examples for which the model at checkpoint 500,000 (text-only epoch) is incorrect and the model at checkpoint 550,000 (image-caption epoch) is correct have the correct sen-

tence of the pair starting with “Is" followed by a noun phrase. For example, pairs such as (“Is the host expecting an award-winning director that hasn’t finished dressing yet?", “Hasn’t the host is expecting an award-winning director that finished dressing yet?"). This contrasts with the distribution of the task, where 31.1% of the pairs have the correct sentence starting with “Is" followed by a noun phrase. We theorise that the Localized Narratives dataset supports the model at checkpoint 550,000 (image-caption epoch) in choosing the "Is" followed by noun phrase sentences with higher probability, which happen to be the correct sentences in these pairs. That is because there are 706,251 constructions of the form “there is" followed by a noun phrase in the Localized Narratives dataset. We hypothesise that as a result, our model learns that the pattern “is" followed by a noun phrase is more likely during the image-caption epochs.

2. For the *turn taking* subtask, even if the model at checkpoint 550,000 (image-caption epoch) chooses the correct sentence more often, it does so with little confidence. For most examples for which checkpoint 550,000 (image-caption epoch) is correct and checkpoint 500,000 (text-only epoch) is not, the log probability difference between the correct and incorrect sentence of the former checkpoint is less than 2 points. To put this in context, the log probability scores range between -89 and -156, for which 2 points represent 0.013% to 0.0225%. There is no noticeable pattern in the training data that can motivate the model’s better performance during image-caption epochs on the the *turn taking* subtask. We conclude that this behaviour requires further investigation which we leave for future work.

The base model achieves better performance on VQA during text-only epochs. In figure 5b, it can be noticed the score of our base model on VQA oscillates by 5% to 10% between text-only epochs and image-caption epochs. We theorise that the cause of these variations is the difference in textual data between the two types of epochs. There are no turn-taking constructions in the image-caption datasets, and the number of questions (25,300 question marks) is significantly lower than in the text-only datasets (1,083,559 question marks). However, both are present in the format of the VQA text data. Therefore, we conclude that the image-caption datasets support the VQA task less due to differences in the text format. We argue that for a high score on VQA during image-caption epochs, the image-caption datasets should contain samples similar to the task.

The alternation between text-only and image-caption epochs has little to no effect on the BLiMP, EWoK and Winoground benchmarks for the base model. As shown in figure 5, there is little oscillation between text-only and image-caption epochs on the BLiMP benchmark, suggesting that the text-only dataset supports the model better for this task, but the score generally increases. There are no noticeable patterns for EWoK or Winoground.

Note: The reason the scores in all benchmarks stabilise after checkpoint 800,000 is because of the small learning rate (5e-5) combined with the learning rate schedule (cosine annealing) we chose for training. After checkpoint 800,000, the learning rate gradually decreases from 1e-5 to 0, which has little effect on the gradients.

G Correlation to Imageability Scores

Category	Gate Selection	
	Mean (SD)	# words
Very Low (<342)	0.427 (0.143)	401
Low (342-450)	0.481 (0.130)	96
High (450-558)	0.378 (0.126)	78
Very High (>558)	0.351 (0.155)	140

Categories defined as $\mu \pm \sigma$ based on the MRC database.
SD = standard deviation.

Table 3: Mean gate value per imageability bin for our base model (incorporating a soft gate per feature).

Table 3 summarises the mean gate values for our base model corresponding to meaningful imageability bins, defined using cutpoints at mean ± 1 SD from the MRC database (Coltheart, 1981).

H Experiments Setup

For all architectural features and training strategies we define in subsections 3.3-E and A-C, we conduct experiments in the form of ablation studies in order to evaluate each potential improvement in isolation. We select our base architecture, described in section 3.2, and define one experiment per feature. We train each enhanced model in the same conditions and evaluate it on five BabyLM Challenge

benchmarks: BLiMP, BLiMP Supplement, EwoK, Winoground and VQA.

H.1 Base Model Implementation Details

We implement our dual stream transformer in PyTorch (Imambi et al., 2021), following the architecture we introduce in section 3.2. We summarise our hyperparameter choices for the base model in table 4.

We use pre-layer normalisation rather than post-layer normalisation in our implementation as previous research shows that pre-layer normalisation provides better training stability for networks larger than six layers (Takase et al., 2022), which is crucial given our limited training budget and inability to perform extensive hyperparameter searches.

Following standard transformer design, we use residual connections around each sub-layer (feed-forward networks, self-attention and cross-attention).

While the image encoder may be over-parameterised for single token processing, empirical results validate this choice (Appendix I), and it ensures architectural consistency and directly comparable results for future extensions to patch-based visual inputs.

H.2 Training Details

We train all of our models using the hyperparameters summarised in table 5 using the BabyLM Challenge 2024 evaluation pipeline, with the exception of a few changes for the auxiliary objective function and data curriculum experiments. In the case of the auxiliary objective function experiments, we increase the batch size from 64 to 128 as a larger batch size is recommended for contrastive learning (Chen et al., 2020), which results in a total of 553,510 steps. Due to computational constraints, we were not able to select a larger batch size. In the case of the data curriculum experiments, the data order differs according to the strategy we define for that experiment. For the model trained using LexiContrastive Grounding as the auxiliary function, we use weight tying as recommended in the original work (Zhuang et al., 2024).

We select a learning rate of 5e-5 to ensure training stability, despite this being conservative for the model size. While alternating between text-only and image-caption epochs improves performance on the benchmarks (as shown in Appendix E), this training regime can cause gradient instability when transitioning between epoch types. Therefore, we adopt a lower learning rate to mitigate this risk.

H.3 Data Pipeline Details

The text-only training dataset is provided in *.txt* files, while the multimodal one is provided in *.json* files for the captions and *.npy* for the image embeddings, where each row in the numpy array embeds one image as a global token of dimension 768. We load the text-only data as one training sample per line, and the image-caption data as one (image, caption) pair representing one training sample. We do not perform any preprocessing on either data.

For the text-only data and the captions, we tokenise the text using the GPT-2 tokeniser (Radford et al., 2019), as our model is autoregressive. We also add the *BOS* and *EOS* special tokens at the beginning and end of each text-only/caption sample, respectively.

We split the data into 80% training, 10% validation and 10% held-out test sets. In order to ensure that all models are trained on the same data, we save the data split indices and reuse them for all experiments. We shuffle the training dataset independently for each run while maintaining consistent train/validation/test partitions.

I Design Choices for the Image Processing Pipeline

Despite using only global image embeddings, we chose to implement an image encoder in our framework for the following reasons:

- Future compatibility: We aim to develop future iterations of this framework that address current limitations by using patch tokens instead of global image embeddings. For comparable results, we choose to use an encoder for the CLS token as well, which benefits from feed-forward and normalisation layers, but not self-attention. The image encoder outputs a non-linear adaptation of pre-trained visual features and improves alignment with the text stream.

- Empirical performance: We experimented with three variants: (1) directly using linearly projected DINOv2 embeddings, (2) applying a 2-layer multi-layer-perceptron (MLP), and (3) using the transformer encoder. The encoder variant demonstrated superior performance across benchmarks, which can be at-

tributed to the encoder's deeper transformation capacity. The benchmark scores for the three variants are available in table 7.

- Computational efficiency: An alternative to the image encoder is to import and fully or partially unfreeze the external pre-trained image encoder used in the BabyLM Challenge, *facebook/dino2-base*[7]. However, this would require processing the raw images through the entire encoder (86.6 million parameters) during training, which would significantly increase the computational costs for data loading, forward passes (and backward passes if unfrozen) and memory usage. This approach contradicts the constraints of the challenge, which advocates for the fair use of computational resources. In contrast, a customisable image encoder component taking as input precomputed embeddings can be modified based on the user's computational constraints.

Table 7 summarises the performance of the base model with different image processing pipelines, evaluated every 200,000 steps on BLiMP, BLiMP Supplement, EWoK, Winoground and VQA using the BabyLM Challenge 2024 evaluation pipeline. As shown, the results on BLiMP, BLiMP Supplement and VQA validate the use of a transformer encoder, for which the base model achieves the best scores. However, since a single image embedding cannot benefit from the self-attention mechanism, an MLP encoder suffices if computational resources are a constraint, achieving competitive performance .Besides the superior performance, the motivation for using a transformer encoder in this work was to enable a direct performance comparison with future iterations of the framework using patch-token embeddings.

[7] https://huggingface.co/facebook/dinov2-base

Model Hyperparameter	Value
Model Dimensions	
Model dimension (d_{model})	768
Hidden dimension	3072
Number of attention heads	8
Image encoder layers	5
Decoder layers	8
Vocabulary & Sequence	
Vocabulary size	50,260 (GPT-2 tokeniser (Radford et al., 2019))
Maximum sequence length	128
Special tokens	[PAD], [BOS], [EOS]
Activation & Regularisation	
Activation function	GELU (Hendrycks and Gimpel, 2016)
Dropout rate	0.1
Layer normalisation	Pre-layer norm
Layer norm epsilon	1e-5 (PyTorch default)
Input Dimensions	
DINOv2 embedding dimension	768
DINOv2 representation	*CLS* token only
Model Statistics	
Total parameters	~198.5M

Table 4: The hyperparameters list for our dual stream transformer base model.

Training Hyperparameter	Value
Data order	Alternating between text-only and image-caption epochs
Number of epochs	10 text-only and 10 image-caption
Total number of steps	1,107,020
Checkpoints saved	Every 50,000 steps
Batch size	64
Learning rate	5e-5
Learning rate schedule	Cosine annealing
Optimiser	AdamW with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$
Number of steps for warmup	~1%
Weight decay	0.01
Gradient clipping norm	1.0
Main loss function	Cross-entropy
Random seed	42

Table 5: The hyperparameters list for our base training regime.

#	Model Architecture	Model Hyperparams	Training Config
1	Base model (§3.2)	Default[a]	Default[b]
Dynamic Gating			
2	Base model + no gate	Default[a]	Default[b]
3	Base model + soft gate per feature	Default[a]	Default[b]
4	Base model + soft gate per token	Default[a]	Default[b]
5	Base model + hard gate per feature	Default[a]	Default[b]
6	Base model + hard gate per token	Default[a]	Default[b]
Feature Representation			
7	Base model + FiLM on text	Default[a]	Default[b]
8	Base model + FiLM on image	Default[a]	Default[b]
9	Base model + FiLM on cross-attention	Default[a]	Default[b]
10	Base model + DyIntra on text	Default[a]	Default[b]
11	Base model + DyIntra on image	Default[a]	Default[b]
12	Base model + DyIntra on cross-attention	Default[a]	Default[b]
13	Base model + Channel Attention	Default[a]	Default[b]
Auxiliary Objectives			
14	Base model	Default[a]	Default[b] + CLIP (BS=128)
15	Base model	Default[a] + weight tying	Default[b] + LCG (BS=128)
Data Curriculum			
16	Base model	Default[a]	Text-only → image-caption[c]
17	Base model	Default[a]	Image-caption → text-only[d]
18	Base model	Default[a]	Non-uniform mix[e]
19	Base model	Default[a]	Uniform mix

[a]As in Table 4 [b]As in Table 5 BS = batch size

[c]First 10 epochs text-only, next 10 epochs image-caption

[d]First 10 epochs image-caption, next 10 epochs text-only

[e]Image-caption and text-only data non-uniformly mixed in same batch

Table 6: Summary of all the experiments we conduct in this work.

Model	Checkpoint	BLiMP	BLiMP S.	EWoK	Winoground	VQA
Base + No Encoder	200K	69.99 ± 0.17	54.03 ± 0.52	49.87 ± 0.57	51.74 ± 1.83	43.00 ± 0.31
	400K	73.21 ± 0.16	53.33 ± 0.62	49.93 ± 0.57	52.68 ± 1.83	46.54 ± 0.31
	600K	72.82 ± 0.16	52.89 ± 0.65	50.38 ± 0.57	52.95 ± 1.83	46.41 ± 0.31
	800K	73.40 ± 0.16	53.69 ± 0.64	50.16 ± 0.57	52.41 ± 1.83	46.52 ± 0.31
	1M	73.17 ± 0.16	53.62 ± 0.64	50.43 ± 0.57	52.14 ± 1.83	46.65 ± 0.31
	1.107M	74.29 ± 0.16	55.63 ± 0.58	50.18 ± 0.57	51.21 ± 1.83	45.02 ± 0.31
Base + MLP Encoder	200K	70.66 ± 0.17	57.42 ± 0.49	50.03 ± 0.57	52.01 ± 1.83	41.72 ± 0.31
	400K	73.47 ± 0.16	51.89 ± 0.66	49.96 ± 0.57	50.67 ± 1.83	48.41 ± 0.31
	600K	73.20 ± 0.16	52.49 ± 0.68	50.02 ± 0.57	52.01 ± 1.83	43.32 ± 0.31
	800K	74.06 ± 0.16	51.71 ± 0.67	50.47 ± 0.57	52.28 ± 1.83	48.18 ± 0.31
	1M	73.71 ± 0.16	50.56 ± 0.67	50.21 ± 0.57	52.55 ± 1.83	48.54 ± 0.31
	1.107M	74.35 ± 0.16	55.38 ± 0.60	50.21 ± 0.57	50.27 ± 1.83	49.83 ± 0.31
Base + Transformer Encoder	200K	69.97 ± 0.17	55.02 ± 0.54	49.83 ± 0.57	51.07 ± 1.83	34.11 ± 0.32
	400K	73.22 ± 0.16	51.88 ± 0.66	50.13 ± 0.57	50.94 ± 1.83	47.82 ± 0.31
	600K	73.47 ± 0.16	53.08 ± 0.62	50.74 ± 0.57	51.61 ± 1.83	48.36 ± 0.31
	800K	74.18 ± 0.16	52.69 ± 0.62	50.69 ± 0.57	52.41 ± 1.83	51.2 ± 0.31
	1M	74.34 ± 0.16	54.00 ± 0.61	50.82 ± 0.57	52.41 ± 1.83	51.2 ± 0.31
	1.107M	75.53 ± 0.16	55.71 ± 0.57	50.41 ± 0.57	51.74 ± 1.83	50.02 ± 0.31

Table 7: The performance of the base model with different image processing pipelines, evaluated using the 2024 BabyLM Challenge evaluation pipeline. The models are evaluated every 200,000 steps on BLiMP, BLiMP Supplement, EWoK, Winoground and VQA.

A Comparison of Elementary Baselines for BabyLM

Rareș Păpușoi and Sergiu Nisioi*
Human Language Technologies Research Center
Faculty of Mathematics and Computer Science
University of Bucharest
rares.papusoi@s.unibuc.ro
sergiu.nisioi@unibuc.ro

Abstract

This paper explores several simple baselines for the BabyLM Challenge, including random models, elementary frequency-based predictors, n-gram language models, LSTMs with various tokenizers (BPE, Unigram, SuperBPE), and GPT-BERT, the winning architecture from the previous BabyLM edition. Evaluation focuses on the BLiMP and BLiMP-Supplement benchmarks. Our experiments reveal that the STRICT-SMALL corpus can sometimes outperform STRICT, that performance is highly sensitive to tokenization, and that data efficiency plays a crucial role. Notably, a simple word-frequency baseline achieved unexpectedly high scores, which led us to identify an evaluation artifact in the pipeline: a system assigning identical sentence-level log-likelihoods to both sentences can attain maximal accuracy. The code for our experiments is available at https://github.com/rarese19/babylm_baselines

1 Introduction

The BabyLM Challenge targets sample-efficient pretraining on developmentally plausible text under strict data budgets (Charpentier et al., 2025). It provides text-only training splits capped at 10M (STRICT-SMALL) and 100M (STRICT) words, drawn from child-directed and conversational sources, with standardized evaluation (Charpentier et al., 2025). In this paper, we operate entirely within the text-only track and treat BabyLM as a fixed environment. Within this setting, we study how model family, tokenizer, and corpus influence grammatical competence under these constraints.

To train our models, we use the official STRICT and STRICT-SMALL corpora, as well as the BabyCosmoFine mixture (Charpentier and Samuel, 2024), which combines equal parts of the BabyLM subset, FineWeb, and Cosmopedia. Evaluation is conducted mostly on BLiMP (Benchmark of Linguistic Minimal Pairs) (Warstadt et al., 2020) and BLiMP-Supplement (Warstadt et al., 2023) from the 2024 evaluation pipeline.

Our results fill a gap in the BabyLM Challenge by providing a comparison between trivial baselines (e.g., random predictions, frequency-based models) and full language models. The contributions of this paper are summarized as follows:

- We present a controlled comparison of model families (n-gram language models, long short-term memory — LSTM, GPT-BERT) and tokenizers (byte-pair encoding, Unigram, SentencePiece, SuperBPE) under fixed BabyLM data budgets and corpora.
- We show that trivial lexical baselines (e.g., raw or Zipf-distributed word frequency) can perform surprisingly well.
- We identify and quantify an evaluator caveat: 22 BLiMP subtasks are permutation-equivalent, allowing order-insensitive systems to tie both sentences. The issue persists on the 2025 evaluation pipeline in a slightly different formulation.
- We establish a strong LSTM (Hochreiter and Schmidhuber, 1997) baseline with 39.2M parameters and analyze tokenizer sensitivity, showing that an 8K vocabulary size with the SuperBPE tokenizer achieves the best performance.
- We train a similar Masked Next Token Prediction model based on slightly altered GPT-BERT recipe (Charpentier and Samuel, 2024). The model achieves the best results and shows that the BabyCosmoFine corpus is better suited for training models on dialogue and question-answering tasks than STRICT-SMALL.

*Corresponding authors.

Proceedings of the First BabyLM Workshop, pages 218–225

Dataset	# Words	
	STRICT	STRICT-SMALL
CHILDES (MacWhinney, 2000)	29M	2.9M
British National Corpus (BNC)	8M	0.8M
Proj. Gutenberg (Gerlach and Font-Clos, 2020)	26M	2.6M
OpenSubtitles (Lison and Tiedemann, 2016)	20M	2.0M
Simple English Wikipedia	15M	1.5M
Switchboard Dialog (Stolcke et al., 2000)	1M	0.1M
Total	**100M**	**10M**

Table 1: Composition of the STRICT and STRICT-SMALL datasets used in BabyLM, adapted from Charpentier et al. (2025).

2 Method

2.1 Datasets

The BabyLM Challenge datasets have only one constraint: they must not surpass 100M words (Charpentier et al., 2025). The competition allows for custom-made datasets as long as they comply with the limitation (Hu et al., 2024; Charpentier et al., 2025). This paper focuses on the STRICT-SMALL track, with a few supplementary experiments conducted under the STRICT category. All experiments use text-only datasets, excluding multimodal and other tracks available in the competition.

The BabyLM Dataset is constructed from multiple text sources in order to create diversity in the language style and the content (Warstadt et al., 2023). It contains text similar to what a child is exposed to during the language acquisition process. The STRICT-SMALL dataset extracts 10% of the 100M words that make up the STRICT corpus and keeps the distribution from the different sources (Warstadt et al., 2023; Charpentier et al., 2025). The dataset's structure is described in Table 1.

The BabyCosmoFine Corpus is created to provide a wider source of information for knowledge extraction and to diversify the language. It consists of a portion of the BabyLM dataset, a portion of the FineWeb-Edu corpus (Penedo et al., 2024), and a portion of the synthetic dataset Cosmopedia (Ben Allal et al.). Each component contributes equally, in terms of quantity, to the overall composition of the corpus (Charpentier and Samuel, 2024).

2.2 Tokenization

We explore a variety of options for data tokenization. Our approach consists of four tokenization schemes, BPE (Gage, 1994; Sennrich et al., 2016), Unigram (Kudo, 2018), their SentencePiece (Kudo and Richardson, 2018) implementations, and SuperBPE (Liu et al., 2025). The tokenizers are trained on both the BabyLM and BabyCosmoFine corpora. BPE incrementally merges frequent adjacent characters to reach a target vocabulary (Sennrich et al., 2016), while Unigram fits a simple probabilistic model over a large possible set and eliminates low-probability sub-words (Kudo, 2018). SentencePiece is an open-source library that provides BPE and Unigram implementations and works directly on raw text without any language-specific pre-tokenization. It treats spaces as a dedicated symbol (e.g., "_"), therefore, segmentation is learned from character sequences instead of using whitespace-defined word boundaries (Kudo and Richardson, 2018).

SuperBPE extends Byte Pair Encoding (BPE) (Liu et al., 2025) with a second training stage that removes whitespace boundaries and learns **super-words** i.e., tokens that can span multiple words (e.g., *by the way*, *I am!*). In the first stage, a standard BPE vocabulary is learned; in the second stage, the learned tokens are re-merged without enforcing spaces as hard boundaries, enabling frequent multi-word expressions to be represented as single tokens. All our SuperBPE tokenizers follow the default configuration, combining 90% regular tokens and 10% super-words.

2.3 Evaluation

Models are evaluated on tasks designed to assess core linguistic abilities and generalization from limited data. The evaluation suite spans grammatical phenomena, general knowledge, information tracking, reading comprehension, and morphological derivation. In this work, we focus on the minimal-pair grammatical acceptability suite, which consists of pairs of nearly identical sentences where the goal is to prefer the grammatical one. These tasks target syntax, morphology, and semantics (e.g., subject–verb agreement, binding).

BLiMP (Benchmark of Linguistic Minimal Pairs) evaluates a model's grammatical competence using sentence pairs that differ in exactly one syntactic, morphological, or semantic feature. Each example contains two sentences (one grammatical and one ungrammatical) and the model must identify the grammatical one by assigning it a higher probability (Warstadt et al., 2020).

BLiMP-Supplement extends BLiMP with examples focused on dialogue and question constructions. This test set was first introduced in the initial edition of the BabyLM Challenge. Its structure mirrors that of BLiMP but targets linguistic phenomena characteristic of conversational language (Warstadt et al., 2023).

The scoring method is based on sentence-level log-likelihood. Autoregressive models are evaluated by summing token-level log-probabilities, whereas masked language models use pseudo log-likelihood, computed by masking each token in turn and summing the resulting log-probabilities. For each minimal pair, the higher-scoring sentence is considered correct, and accuracy is then aggregated across subtasks.

2.4 Approaches

We consider simple baselines, alongside the ones provided by the competition (Choshen et al., 2024; Charpentier et al., 2025), classical n-grams, a recurrent model, and transformers, all within the BabyLM constraints.[1]

The controlled-random baseline sets a target sentence log-probability $R \sim \mathcal{U}[-100, 0]$ and returns the same logit vector at every position, independent of the input. Let L be the number of scored tokens and V the vocabulary size. We choose a constant logit α so that each reference token has probability $p = \frac{e^\alpha}{e^\alpha + (V-1)}$, which yields a sentence score

$$L\big[\alpha - \ln\big(e^\alpha + V - 1\big)\big] = R \tag{1}$$

We solve for α via a short binary search and then output the same logit vector at every position, α on the reference token at that position and 0 on all others, independent of the input.

The word frequency baseline assigns a sentence score by summing each word's relative frequency in a reference corpus. We use a frequency table based on each sub-corpus. For a sentence x,

$$S(x) = \sum_{w \in x} \mathrm{freq}(w) \tag{2}$$

with

$$\mathrm{freq}(w) = \begin{cases} \dfrac{c(w)}{N}, & c(w) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $c(w)$ is the corpus count of w and N is the total token count. This score ignores word order and context.

As an alternative, we use a Zipf-style score inspired by `wordfreq` (Speer, 2022; Van Heuven et al., 2014):

$$\mathrm{zipf}(w) = \begin{cases} 3 + \log_{10}\big(10^6 c(w)/N\big), & c(w) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The factor 10^6 scales to "per million" and the constant 3 keeps values positive.

N-gram language models are trained with KenLM (Heafield, 2011) for orders $n \in \{2, \ldots, 6\}$ on the STRICT and STRICT-SMALL corpora. Training uses sentence-boundary markers, and a single <unk> token for out-of-vocabulary items; the tokenization is done with the default configuration of KenLM. Models are compiled to KenLM binaries and evaluated by the BabyLM evaluation pipeline, identical to the neural models.

Long Short-Term Memory (LSTM) is a neural language model (Hochreiter and Schmidhuber, 1997) with 39.2 million parameters. We train the model on the STRICT and STRICT-SMALL splits, using the tokenizers described in Section 2.2 (BPE, Unigram, SuperBPE). Each configuration is trained for 10 epochs with identical hyperparameters across corpora, and no external text is used.

Transformer language models (Vaswani et al., 2017) are treated as standard modeling tools, with our focus placed primarily on training objectives and configurations. GPT-BERT, the winning model from the 2024 BabyLM Challenge, employs Masked Next Token Prediction (MNTP) alongside a causal language modeling objective (BehnamGhader et al.; Charpentier and Samuel, 2024). We follow the publicly released training recipe, with using the script designed for single GPU training.[2] The script schedules late-training changes at 70% and 90% of steps; due to resource constraints we apply only the $\geq 70\%$ change and

[1] The 2025 BabyLM rules cap training at *10 epochs*. Some runs in this paper do not comply: they exceed 10 epochs due to the project timeline and because parts of the work predated this year's rules. We report these results for analysis, not as an official challenge submission, while still respecting the 10M/100M word data limits.

[2] The single-GPU script has been removed in the meantime from the official release of GPT-BERT because it was not equivalent to the original multi-GPU training.

omit the $\geq 90\%$ change. MNTP masks a subset of input tokens and learns to predict each masked token using the hidden state at the preceding position.

Config.	BLiMP	BLiMP-Supp.
STRICT$_{\textit{rel. frequency}}$	0.653	0.637
STRICT$_{\textit{Zipf}}$	0.661	0.658
STRICT-SMALL$_{\textit{rel. frequency}}$	0.654	0.642
STRICT-SMALL$_{\textit{Zipf}}$	**0.663**	**0.661**

Table 2: Performance of the word-frequency scoring method on the STRICT and STRICT-SMALL corpora. The Zipf variant slightly outperforms relative frequency on both datasets.

3 Experiments

Unless noted, all scores in this section use the 2024 evaluator (Warstadt et al., 2023). The controlled-random baseline reaches **0.543** on BLiMP and **0.430** on BLiMP-Supplement, which serves as a true no-signal floor.

The results in Table 2 show that the word-frequency baseline is unexpectedly strong despite ignoring order and context. A Zipf weighting consistently outperforms relative frequencies, likely because the logarithmic scale better matches the log-likelihood scoring in the evaluator. Scores are essentially unchanged between STRICT and STRICT-SMALL, implying that corpus size contributes little once unigram statistics are learned.

N-gram language models (Table 3) are below the word-frequency baseline on both corpora. Accuracy increases with n and then plateaus, consistent with gains from local collocations rather than deeper structure. Notably, STRICT-SMALL often outperforms STRICT, suggesting that its distribution overlaps more with the linguistic patterns probed by the evaluation, despite its smaller size.

The LSTM models appear to be highly tokenizer-sensitive (Table 4). SuperBPE[3] shifts the distribution of units toward multi-word chunks, which helps slightly on dialogue/question phenomena but does not consistently improve core grammatical judgments. The results in Table 4 are directly comparable to the BabyHGRN (Haller et al., 2024) setup on STRICT-SMALL. BabyHGRN benchmarks sub-quadratic recurrent networks under the same BabyLM data budgets and includes an LSTM baseline. Our results show that tokenizer choice and vocabulary size affect accuracy on STRICT-

[3] https://huggingface.co/UW/OLMo2-8B-SuperBPE-t180k

KenLM	BLiMP	BLiMP-Supplement
2-gram$_{\textit{Strict}}$	0.596	0.552
2-gram$_{\textit{Strict-Small}}$	0.627	0.589
3-gram$_{\textit{Strict}}$	0.592	0.562
3-gram$_{\textit{Strict-Small}}$	0.632	0.587
4-gram$_{\textit{Strict}}$	0.598	0.572
4-gram$_{\textit{Strict-Small}}$	0.634	0.596
5-gram$_{\textit{Strict}}$	0.598	0.569
5-gram$_{\textit{Strict-Small}}$	0.634	0.603
6-gram$_{\textit{Strict}}$	0.598	0.570
6-gram$_{\textit{Strict-Small}}$	**0.633**	**0.606**

Table 3: Performance of KenLM n-gram models trained on the STRICT and STRICT-SMALL corpora, evaluated on BLiMP and BLiMP-Supplement. Despite the smaller data size, STRICT-SMALL often yields higher scores.

Vocab.	Tokenizer	BLiMP	BLiMP-Supp.
4k	SentencePiece BPE	0.644	**0.555**
	SentencePiece Unigram	0.646	0.547
	SuperBPE (trained)	**0.657**	0.536
8k	SentencePiece BPE	0.640	**0.581**
	SentencePiece Unigram	0.630	0.514
	SuperBPE (trained)	**0.661**	0.553
16k	SentencePiece BPE	0.607	0.522
	SentencePiece Unigram	**0.646**	0.537
	SuperBPE (trained)	0.613	**0.550**
	SuperBPE (pretrained)†	0.637	0.551

Table 4: LSTM performance on STRICT-SMALL grouped by tokenizer vocabulary size. The models are trained on the BabyLM dataset. Mid-size vocabularies (8k) yield the best BLiMP (SuperBPE–8k) and BLiMP-Supplement (BPE–8k), while Unigram is strongest at 4k; overall, tokenizer choice impacts accuracy more than vocabulary size. †Uses an externally pretrained vocabulary.

Dataset	Tokenizer	BLiMP	BLiMP-Supp.
Strict-Small	BPE	0.794	0.591
	Unigram	**0.796**	0.633
	SuperBPE	0.787	0.588
BabyCosmoFine	BPE	0.791	0.705
	Unigram	0.801	**0.715**
	SuperBPE	**0.803**	0.692

Table 5: 8k vocab GPT-BERT performance on STRICT-SMALL and BabyCosmoFine across tokenizers, evaluated on BLiMP and BLiMP-Supplement. Models trained on BabyCosmoFine score higher on BLiMP-Supplement, indicating better coverage of dialogue/question phenomena.

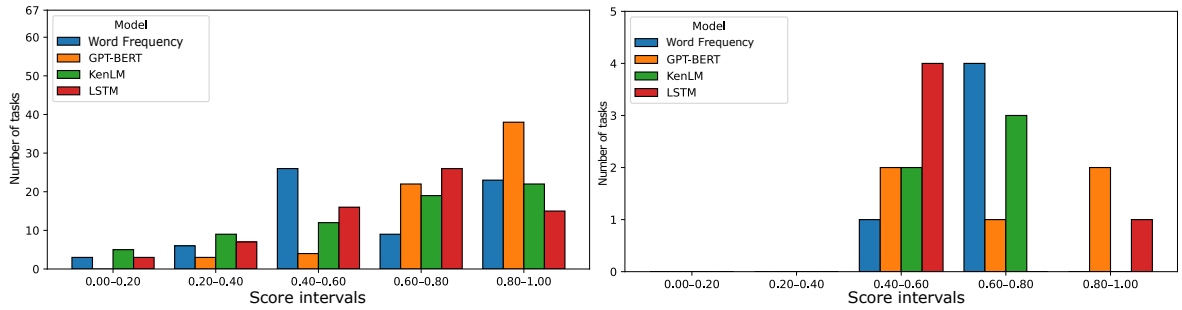


Figure 1: On the left, the distribution of model scores across BLiMP subtasks. On the right, the distribution of model scores across BLiMP-Supplement subtasks. GPT-BERT's scores cluster in the upper ranges, whereas the other models show a wider spread in performance. Only GPT-BERT and the LSTM achieve high scores for BLiMP-Supplement, showing how challenging these tasks can be.

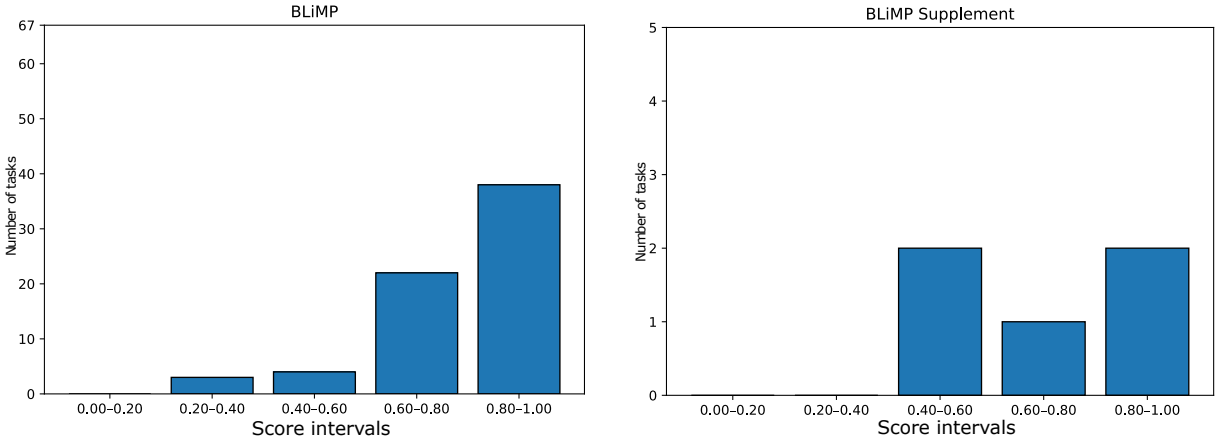


Figure 2: Distribution of GPT-BERT scores across BLiMP and BLiMP-Supplement subtasks. The share of BLiMP-Supplement subtasks with high scores is smaller than in BLiMP, indicating weaker performance on dialogue and question examples.

SMALL. The strongest BLiMP score is SuperBPE-8k, while BLiMP-Supplement is best with BPE-8k. Unigram is competitive at 4k but degrades at 8k. SuperBPE benefits from mid-size vocabularies on BLiMP and from larger vocabularies on BLiMP-Supplement.

Two consistent trends emerge: mid-size vocabularies favor dialogue/question phenomena across tokenizers, most clearly for BPE, while smaller vocabularies help Unigram on BLiMP. Overall, tokenizer family matters more than vocabulary size for LSTM models.

GPT-BERT (Charpentier and Samuel, 2024) is the strongest model in this study (see Table 5). Training it on BabyCosmoFine yields a clear lift on dialogue/question phenomena compared to STRICT-SMALL, pointing to a domain-coverage effect rather than purely scaling with data volume. Tokenizer choice matters less than for the LSTM; we treat GPT-BERT$_{(Unigram,\ BabyCosmoFine)}$ as the reference configuration.

4 Results Analysis

Figure 1 contains the distribution of BLiMP subtask accuracies for all systems. The results show that GPT-BERT clusters near the top, the LSTM sits in the mid–high range, and classical/lexical baselines spread widely. The pattern suggests that much of BLiMP can be addressed by stronger modeling capacity on top of lexical priors, with diminishing returns from short-context statistics alone. For the BLiMP-Supplement tasks, the distributions shift downward. Only GPT-BERT and the LSTM place substantial mass in the higher bins, underscoring the difficulty these tasks pose for most systems. This supports the view that BLiMP-Supplement probes conversational structures and question forms that benefit from models trained on dialogue-oriented corpora (e.g., BabyCosmoFine) and from tokenization schemes that stabilize sequence modeling. Figure 2 shows that GPT-BERT scores are concentrated in the upper bins for BLiMP, while BLiMP-Supplement is flat-

ter with fewer high-scoring subtasks. This gap mirrors our earlier results: core grammatical minimal pairs are largely handled, whereas dialog/question phenomena remain uneven, pointing to a domain-coverage effect rather than pure data scaling.

Table 6 shows GPT-BERT's strongest BLiMP subtasks: `irregular_past_participle_adjectives`, `determiner_noun_agreement_2`, and `determiner_noun_agreement_1`, all targeting morphology. Among the other systems, only the LSTM remains consistently competitive across these three; the word-frequency baseline does not stand out, and KenLM comes close on `determiner_noun_agreement_1`, consistent with short-range determiner–noun collocations. Representative items and each model's choice are given in Table 7.

Table 8 illustrates three BLiMP subtasks on which GPT-BERT performs worst. While the word-frequency baseline sometimes appears to answer these items correctly, this is largely an evaluation artifact. In 22 of the 67 BLiMP subtasks, the two sentences in each minimal pair are permutations of the same multiset of words. Any scorer that is invariant to word order assigns identical sentence scores to both sides of such pairs.

In the 2024 evaluator, ties are counted as correct, which inflates accuracy for permutation-equivalent items. When we exclude these 22 subtasks, the Zipf word-frequency mean drops from 0.663 to 0.498 on STRICT-SMALL, confirming that the initial score was driven by the artifact rather than genuine grammatical competence.

In the 2025 evaluator, identical sentence scores are still marked as correct by the evaluator. When the two candidates in a BLiMP minimal pair obtain the same sentence-level log-likelihood, the item is counted as correct. We construct a dummy model that assigns the same log-likelihood to both candidates in the benchmark. Concretely, at each scored position, we set the observed next-token logit to 0 and all other vocabulary logits to $-\infty$. The evaluator computes per-token log-probabilities with `log_softmax` and then sums only positions selected by the phrase mask. If a masked position leaves no valid entry (i.e., the row is all $-\infty$), `log_softmax` returns NaN. These NaNs propagate to both candidates' sentence totals, making them numerically indistinguishable; the evaluator treats this as a tie and marks the item correct. In practice, this yields a reported score of 100.0. For completeness, we also created a finite-negative variant that avoids NaNs by setting 0 on target tokens and -K on others, which yields a near-perfect score (99.69). Because non-target logits are finite, positions where the phrase mask removes the reference token contribute a constant offset $(-\log V)$ rather than 0. Minimal pairs can differ in how many such positions they contain, so the two sentence totals are not exactly equal; a small fraction of items cease to be ties, therefore the almost-perfect score.

5 Conclusions

We examine data-efficient pretraining in the BabyLM setting across classical and neural families while and varying tokenizer and corpus. Word–frequency signals already go far on BLiMP, obtaining scores as high as 0.66, exceeding LSTMs and n-gram language models. The frequency baseline achieves a 25% decrease in score apparent after removing the 22 permutation-based subtasks, where short-range collocations help. For n-gram language models STRICT-SMALL is a better choice than STRICT indicating that arbitrary changes in the dataset can have an impact of at least 0.05 in evaluation scores. This raises an important concern on when to decide if a system is actually stronger than another. The LSTM is sensitive to tokenization and GPT-BERT is the strongest model. Regarding corpus effects, we evaluated BabyCosmoFine only with GPT-BERT; in that setting, it yields clear improvements on dialogue and question-related phenomena compared to STRICT-SMALL.

Some BLiMP subtasks contain sentence pairs that are permutations of the same words. This led us to discover that, in both the 2024 and 2025 evaluators, when the two candidates receive the same sentence-level log-likelihood, the item is counted as correct; consequently, a no-signal system that enforces equal scores can report a score of 100.0.

Overall, tokenizer and data choices are relevant factors alongside model family at the 10M-word scale, and reporting across tokenizers helps make comparisons more informative.

Acknowledgments

This work was supported by the Romanian National Research Council (CNCS) through the Executive Agency for Higher Education, Research, Development and Innovation Funding (UEFISCDI) under grant PN-IV-P2-2.1-TE-2023-2007 InstRead.

BLiMP subtask	Description	GPT-BERT	WordFreq	KenLM	LSTM
irregular_past_participle_adjectives	Use of irregular past participles (e.g., *broken*, *hidden*) as adjectives.	0.996	0.612	0.622	0.874
determiner_noun_agreement_2	Number agreement between determiners and irregular nouns (e.g., *these geese* vs. *this geese*).	0.986	0.495	0.492	0.781
determiner_noun_agreement_1	Number agreement between determiners and regular nouns (e.g., *these dogs* vs. *this dogs*).	0.978	0.496	0.934	0.833
existential_there_quantifiers_2	Existential *there* with quantifiers and regular nouns (e.g., *there was every fish*).	0.236	1.000	0.667	0.418
left_branch_island_echo_question	Left-branch extraction constraint in echo questions (e.g., *Sara was insulting what student?*).	0.337	1.000	0.821	0.702
sentential_subject_island	Extraction from a sentential subject.	0.364	1.000	0.263	0.389

Table 6: GPT-BERT's best and worst BLiMP subtasks, compared with other systems.

BLiMP subtask	Sentences	GPT-BERT	WordFreq	KenLM	LSTM
irregular_past_participle_adjectives	Good: *The worn jacket was smooth.* Bad: *The wore jacket was smooth.*	Correct	Incorrect	Incorrect	Incorrect
determiner_noun_agreement_2	Good: *Robert hates that dancer.* Bad: *Robert hates those dancer.*	Correct	Correct	Correct	Correct
determiner_noun_agreement_1	Good: *Most waiters could break those couches.* Bad: *Most waiters could break those couch.*	Incorrect	Incorrect	Correct	Correct

Table 7: Examples from the three BLiMP subtasks on which GPT-BERT is strongest, showing each model's decision (Correct/Incorrect).

BLiMP subtask	Sentences	GPT-BERT	WordFreq	KenLM	LSTM
existential_there_quantifiers_2	Good: *All students weren't there noticing some box.* Bad: *There weren't all students noticing some box.*	Incorrect	Correct	Incorrect	Correct
left_branch_island_echo_question	Good: *Roger has noticed whose rivers?* Bad: *Whose has Roger noticed rivers?*	Incorrect	Correct	Correct	Correct
sentential_subject_island	Good: *Who would all cars' hurting Irene bore.* Bad: *Who would all cars' hurting bore Irene.*	Correct	Correct	Incorrect	Incorrect

Table 8: Examples from three BLiMP subtasks on which GPT-BERT is weakest, showing each model's decision (Correct/Incorrect). Although the word-frequency baseline sometimes appears correct on these items, we argue this reflects an evaluation artifact.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Smollm-corpus. https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus. Accessed: 2025-06-06.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [call for papers] the 2nd BabyLM challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2404.06214.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Patrick Haller, Jonas Golde, and Alan Akbik. 2024. BabyHGRN: Exploring RNNs for sample-efficient language modeling. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural*

Language Learning, pages 82–94, Miami, FL, USA. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. *Preprint*, arXiv:2412.05149.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. Extracting large parallel corpora from movie and tv subtitles. Technical Report 2016:22, University of Oslo.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. 2025. Superbpe: Space travel for language models. *Preprint*, arXiv:2503.13423.

Brian MacWhinney. 2000. *The CHILDES Project: The Database*, 2 edition, volume 2 of *Tools for Analyzing Talk*. Psychology Press, Mahwah, NJ.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, and 1 others. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Robyn Speer. 2022. rspeer/wordfreq: v3.0 (v3.0.2).

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Walter JB Van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. Subtlex-uk: A new and improved word frequency database for british english. *Quarterly journal of experimental psychology*, 67(6):1176–1190.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Two ways into the hall of mirrors: Language exposure and lossy memory drive cross-linguistic grammaticality illusions in language models

Kate McCurdy and Katharina Christian and Amelie Seyfriend and Mikhail Sonkin
Saarland University

Abstract

Readers of English — but not Dutch or German — consistently show a *grammaticality illusion*: they find ungrammatical double-center-embedded sentences easier to process than corresponding grammatical sentences. If pretrained language model (LM) surprisal mimics these cross-linguistic patterns, this implies that language statistics explain the effect; if, however, the illusion requires memory constraints such as lossy context surprisal (LCS), this suggests a critical role for memory. We evaluate LMs in Dutch, German, and English. We find that both factors influence LMs' susceptibility to grammaticality illusions, and neither fully account for human-like processing patterns.

1 Introduction

Modern neural language models (LMs) produce fluent, grammatical language (Mahowald et al., 2024), but their validity as models of human linguistic cognition remains contested (Cuskley et al., 2024). One key concern is the scale of data exposure: LMs often learn from quantities of linguistic data which exceed human lifespans. This motivates research with LMs trained on human-scale data, as these models may have a greater claim to cognitive plausibility (Wilcox et al., 2025).

Another dimension of cognitive plausibility concerns language processing rather than learning. Language model surprisal robustly predicts measures of incremental language processing such as reading time (Wilcox et al., 2023). Despite this, LMs fail to fully reproduce certain processing effects which are well-established in the experimental literature, such as recovery from syntactically ambiguous "garden-path" sentences (Arehalli et al., 2022; Huang et al., 2024). Such systematic divergences from human processing further challenge LMs' cognitive plausibility. Moreover, the two issues may be connected: larger LMs trained on more data are worse at approximating reading time (Oh and Schuler, 2023), raising the possibility that non-human-scale learning contributes to non-human-like processing.

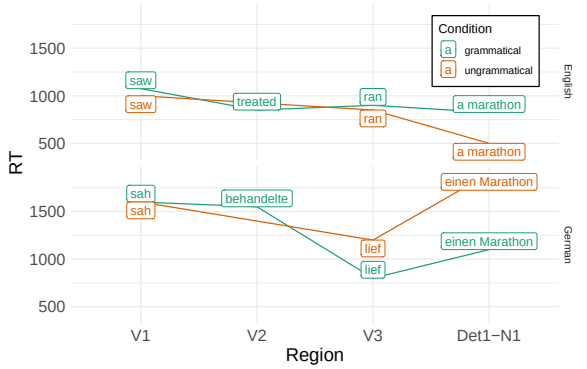


Figure 1: Example of the grammaticality illusion in reading time (RT) for (1a) vs. (1b) in English (upper), and the converse effect in German (lower), using mean RTs from Vasishth et al. (2010, Expts. 1 and 3). German speakers take longer to read the post-verbal NP ("einen Marathon") in the ungrammatical, missing-verb condition, while English speakers instead read the NP faster when the verb is missing.

One important lens on linguistic cognition comes from language *illusions*, i.e. ungrammatical or otherwise infelicitous sentences which humans should reject, but nonetheless find acceptable (Phillips et al., 2011). For example, consider the following two sentences (cf. Figure 1):

1. (a) *The painter who the doctor who the lady saw treated ran a marathon.*
   (b) **The painter who the doctor who the lady saw ran a marathon.*

(1a) contains double nested center-embedded clauses, which are rare and challenging in English (Hamilton and Deese, 1971) but indisputably grammatical. (1b), on the other hand, is ungrammatical due to missing a verb. Despite this, speakers consistently prefer sentences like (1b) to their grammatical counterparts (Gibson and Thomas, 1999;

Proceedings of the First BabyLM Workshop, pages 226–236

Christiansen and MacDonald, 2009). This processing effect is known as the *grammaticality illusion*.

What causes the grammaticality illusion? Two competing hypotheses have been proposed. Under the *memory* account, constrained working memory causes readers to forget earlier sentence material, thereby nullifying the expectation of a third verb (Gibson and Thomas, 1999). This proposal appeals to general cognitive mechanisms; however, Vasishth et al. (2010) and Frank et al. (2016) find that the illusion appears for reading times in English, but not in German or Dutch. They posit an alternative *language statistics* hypothesis: the grammaticality illusion arises due to the relative rarity of center-embedded clauses in English. Researchers have evaluated these two accounts with computational models (Engelmann and Vasishth, 2009; Christiansen and MacDonald, 2009; Frank, 2014; Futrell et al., 2020). These simulations, however, predate today's language models (LMs), which represent linguistic distributions to unprecedented levels of precision.

In this paper, we use modern LMs to assess these two hypotheses with respect to the grammaticality illusion. LMs effectively implement the language statistics account. If the distribution of English gives rise to the illusion, then English LMs should assign higher surprisal to the post-verbal region *a marathon* in the grammatical sentence (1a) compared to the ungrammatical (1b), while German and Dutch LMs should do the converse. Moreover, this cross-linguistic divergence should hold steady, or even grow, with increased training data: if language statistics drive the effect, then higher data exposure during training should reinforce these respective language-specific outcomes.

If, however, memory constraints are critical to the illusion, we may see two different patterns. Firstly, the grammaticality illusion in English may be mediated by language model capacity in the *opposite* direction (cf. Oh and Schuler, 2023). In this scenario, smaller models trained on human-scale data may show the illusion, while larger LMs consistently prefer the grammatical sequence. Secondly, the grammaticality illusion may be mediated by retention of the preceding linguistic context, such that Lossy Context Surprisal shows the effect at higher forgetting rates (LCS; Futrell et al., 2020).

Our results suggest that both language statistics and memory constraints influence how LMs process double-center-embedded sentences, with mixed implications for human sentence processing. For the language statistics hypothesis, we find robust support from Dutch, but for German and English the picture is more complicated. Larger English LMs reproduce the missing verb illusion, which decisively supports the hypothesis. Medium-sized models, however, show the opposite effect, and larger German LMs also unexpectedly show the illusion — neither of which are expected under a language statistics account. For the memory hypothesis, we find that Resource-Rational Lossy Context Surprisal (RR-LCS; Hahn et al., 2022) simulates the grammaticality illusion at higher forgetting rates in both English and German; however, we do not observe the expected language-specific interaction in effect direction. These findings highlight continuing challenges in applying LMs as cognitive models of human language processing.

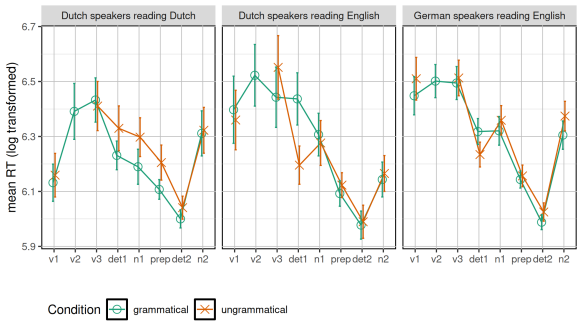


Figure 2: Grammaticality effects in reading time (RT) for Dutch and English stimuli, reproduced from Frank et al. (2016, Expts. 1–3). At the post-verbal determiner (Det1), RT is higher for ungrammatical sentences in Dutch (left), but lower in English (middle, right).

2 Related work

2.1 Grammaticality illusion effects across languages

English speakers consistently prefer ungrammatical sentences like 1b to grammatical double-center-embedded sentences like 1a. This missing verb effect has been found in both acceptability judgments (Gibson and Thomas, 1999; Frank and Ernst, 2019; Huang and Phillips, 2021) and measures of online processing such as reading time (Vasishth et al., 2010; Frank et al., 2016, 2021). Some researchers have argued that this effect reflects language-specific distributions of center-embedded sentences (e.g. Vasishth et al., 2010; Pañeda and Lago, 2024). Figure 2 presents key evidence for this **language statistics** hypothesis: the missing verb effect appears in English, but not in the related

language Dutch. Strikingly, the English reading times shown in Figure 2 come from first language speakers of Dutch and German — speakers who do not show this effect when reading their native language. The fact that the grammaticality illusion appears robustly for particular *languages*, rather than particular *speakers*, supports the language statistics hypothesis. Under this account, we would expect modern language models to show comparable language-dependent preferences: lower surprisal for grammatical Dutch and German sentences, and ungrammatical English sentences.

The main alternative hypothesis states that the grammaticality illusion is driven by domain-general **memory constraints** (Gibson and Thomas, 1999). Language-specific effects (cf. Figure 2) clearly challenge this account. In response, some recent work has explored how working memory could be mediated by particular linguistic properties. For instance, relative clauses in German and Dutch use different word order from main clauses; Bader and colleagues (Bader, 2016; Häussler and Bader, 2015) argue that this syntactic distinction may facilitate retrieval from memory in German, but not English. Huang and Phillips (2021) build on this account to characterize a similar illusion in Mandarin. Finally, Futrell and colleagues (Futrell and Levy, 2017; Futrell et al., 2020) integrate the two hypotheses with a model of memory directly mediated by language statistics, which we consider at length in the following section.

2.2 Processing illusions in language models

A number of computational modeling studies have found support for the language statistics hypothesis, although much of this previous work relies upon simulated training data. Christiansen and Chater (1999), Engelmann and Vasishth (2009), and Christiansen and MacDonald (2009) trained recurrent neural networks (RNNs) on two distinct probabilistic context-free grammar (PCFG)-generated corpora with differing relative clause distributions, reflecting corpus frequencies from German and English. These models capture human-like grammaticality preferences for double center-embedded constructions in each language respectively. Frank (2014) trained RNN language models on natural corpora from English and Dutch, and Frank et al. (2016) find that these models reproduce the language-specific grammaticality effects observed in their behavioral experiments (Figure 2). Notably, however, the English model did not fully reproduce the strength of the preference for ungrammatical sentences; it showed lower surprisal in the missing verb condition, but this difference did not reach statistical significance.

Futrell et al. (2020) introduce lossy context surprisal (LCS), a model which synthesizes memory-based and language statistics accounts. The core intuition is that speakers rely on noisy memory representations to predict upcoming words, and the noisy memory is more likely to recall structures which are more frequent in their language. Therefore, even if speakers typically forget 20–30% of the words in a given sentence context, a German speaker is more likely to correctly retain multiple verb-final relative clauses than an English speaker, simply due to the greater prevalence of such constructions in German. Futrell et al. similarly evaluate their model on a PCFG-derived corpus, with the additional manipulation of a forgetting parameter. Their LCS model predicts that, at certain levels of memory loss, English comprehenders will exhibit the grammaticality illusion, whereas German comprehenders will not. The LCS model thus provides, in principle, a memory-based account for language-specific grammaticality effects (although cf. Huang and Phillips, 2021).

If the grammaticality illusion in English reflects memory constraints, however, we would not expect it to arise from modern Transformer-based language models (LMs) (Vaswani et al., 2017), which have drastically larger memory capacities than the RNNs evaluated by Frank et al. (2016). Modern LMs have shown some susceptibility to other language illusions, for instance in processing negative polarity items (Zhang et al., 2023) and number agreement (Arehalli and Linzen, 2024). In terms of memory, however, modern LMs seem to show superhuman linguistic memory in certain respects (Oh and Schuler, 2023; Oh et al., 2024). The missing verb illusion, then, presents a key test case for the language statistics hypothesis: if it truly reflects language statistics rather than memory limitations, then large modern LMs should reproduce the preference for ungrammatical double center-embeddings in English.

3 Methods

3.1 Models and measures

Surprisal We test[2] multiple pretrained language models (LMs), listed in Table 1.We select these

[2] github.com/kmccurdy/grammaticality-illusion-LMs

Language	Model	Family	Parameters	Training data	Reference
Dutch	GPT2-S Dutch	GPT	129M	33B	de Vries and Nissim (2020)
	GPT2-M Dutch	GPT	380M		
	LLaMA2 Dutch	Llama	13B	2B	Vanroy (2023)
German	GerPT2-L	GPT	876M	50B	Minixhofer (2020)
	BLOOM	GPT	6.4B	50B	Ostendorff and Rehm (2023)
	LEO-LM	Llama	7B	65B	Plüster and Schuhmann (2023)
			13B		
English	GPT-BERT-S	GPT-BERT	30M	100M	Charpentier and Samuel (2024)
	GPT-BERT	GPT-BERT	119M		
	GPT2-mini	GPT	39M	≈2.25B	Fagnou (2024)
	GPT2	GPT	137M	≈15B	Radford et al. (2019)
	GPT2-L	GPT	812M		
	GPT-Neo	GPT	2.72B	420B	Black et al. (2021)
	GPT-J	GPT	6B		
	DeepSeek[1]	DeepSeek	7B	2T	DeepSeek-AI et al. (2024)
	LLaMA2	Llama	7B	2T	Touvron et al. (2023)
			13B		
Multiple	mGPT	GPT	1.3B	≈450B EN, 100B DE, 50B NL	Shliazhko et al. (2024)
			13B		
	LLaMAX	Llama	6.74B	≈950M EN, 900M DE, 590M NL	Lu et al. (2024)
	EuroLLM	Llama	9.15B	≈2T EN, 240B DE, 100B NL	Martins et al. (2024)

Table 1: Language models used in experiments. 'Training Data' refers to language-specific training data in tokens. Note that all monolingual Dutch and German models are initialized from models pre-trained on English.

models to span a range of sizes and training regimes, but focus only on models trained on the language modeling objective, e.g. excluding instruction-tuned models.

We follow previous work (e.g. Futrell et al., 2019) in using LM surprisal to measure incremental processing difficulty. Surprisal is calculated as the negative log probability of a word[3] w_T conditioned on the sequence of preceding words:

$$-\log P(w_{T+1}|w_{1...T}) \quad (1)$$

Lossy Context Surprisal Lossy context surprisal (Futrell et al., 2020) has been proposed to model language-specific effects of constrained memory. We use a specific implementation: resource-rational lossy context surprisal (RR-LCS), proposed by Hahn et al. (2022). RR-LCS provides a fully data-driven implementation of LCS, with only one free parameter: the forgetting rate. Crucially, we can train a range of individual RR-LCS model instances at different forgetting rates to simulate different patterns of working memory engagement. As all aspects of the RR-LCS model are learned from monolingual corpora, we expect that this model is capable of learning and reproducing language-specific effects.

In contrast to standard surprisal, which conditions on an exact word sequence, LCS conditions on a noisy memory representation of the preceding context:

$$-\log P(w_{T+1}|M_T) \quad (2)$$

where M is a lossy representation generated from $w_1 \dots w_T$. At a given forgetting rate, for a given word sequence $w_{1...T}$, RR-LCS learns to stochastically retain or delete specific words from M_T. Reconstructions of the missing words are then sampled, on the basis of language statistics, from a reconstruction model, and the overall surprisal of the noisy sequence is computed using a standard pretrained language model. We refer the reader to Hahn et al. (2022) for further details. As RR-LCS is computationally expensive, we train a limited set of models for two languages, English and German,

[3]Modern LMs are typically trained on a vocabulary of subword tokens rather than words; however, this does not affect our analysis for reasons discussed in the following section.

with the subword version of RR-LCS (McCurdy and Hahn, 2024). We use BLOOM as the base LM for German, and GPT2-L as the base LM for English. We train 3 instances of the RR-LCS model at forgetting rates 20%, 30%, and so on, up to 80%.[4]

3.2 Evaluation

Stimuli We evaluate the grammaticality illusion using the Dutch, German, and English stimuli developed by Vasishth and colleagues (Vasishth et al., 2010; Frank et al., 2016). Each stimulus item appears both as a grammatical double-center-embedded construction (e.g. 1a) and with an ungrammatical missing verb (e.g. 1b). Table 2 illustrates an additional manipulation: in the English and German stimuli, subject noun phrases are either all animate, or the second noun is replaced with an inanimate object. Vasishth et al. (2010) describe this manipulation as motivated by *interference*, but do not discuss it any further. Frank et al. (2021) do not reproduce this animacy manipulation, so the Dutch stimuli only include animate subject nouns.

Critical region We focus on the determiner immediately following the third verb, for reasons illustrated by Figure 2. Across all three languages, we observe the grammaticality effect (in German and Dutch) or illusion (in English) on the post-verbal noun phrase, especially the beginning of the phrase — the determiner. This makes sense: if the reader expects a third verb, and sees a determiner instead, this mismatch in expectation should yield higher RT at this location. We also observe higher RT on the noun, but this may reflect spillover effects (e.g. Rayner, 1998). Processing difficulty appears initially on the determiner; therefore, our analysis of LM surprisal focuses on this region.

4 Results

4.1 Language model surprisal

Figure 3 shows language model (LM) surprisal results Dutch and German, including only stimuli in the animate condition. Across model sizes, Dutch LMs show a robust preference for grammatical sentences, reproducing the effect found in human reading times (Frank et al., 2016; Frank and Ernst, 2019). By contrast, larger German LMs show higher surprisal for grammatical compared to ungrammatical stimuli — in other words, they

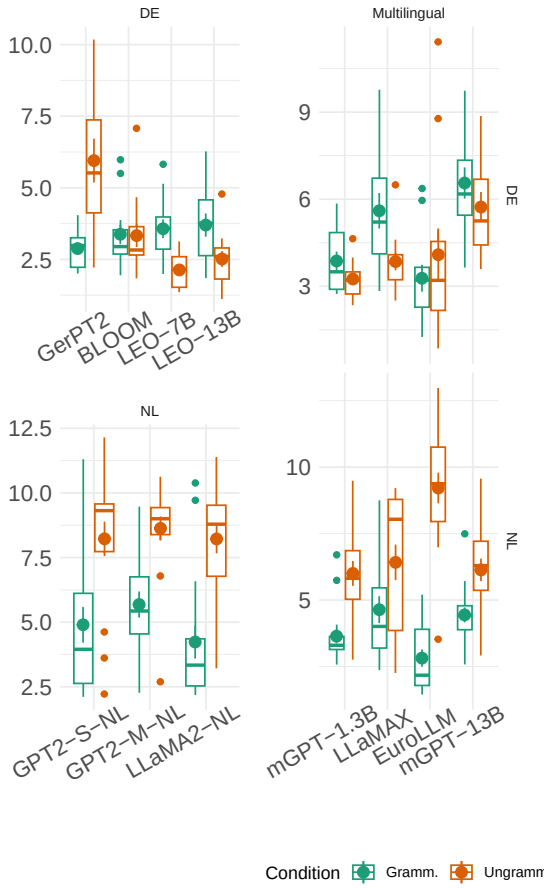


Figure 3: Language model surprisal for animate stimuli in German (upper) and Dutch (lower), for monolingual (left) and multilingual (right) models. Models are ordered left-to-right by parameter count. Dutch models consistently prefer grammatical sentences (lower surprisal), while larger German models unexpectedly show the grammaticality illusion.

show the grammaticality illusion. This is highly unexpected, as Vasishth et al. (2010) conducted multiple experiments with German speakers across different modalities, and never found a preference for ungrammatical sentences.

While Dutch LMs of all capacities prefer grammatical sentences, and German LMs flip from the grammatical to ungrammatical preference as the models grow in size, English language models (Figure 4) show an even more variable trajectory. Small models trained on human-scale data, such as GPT-BERT (the top performing model in the 2024 BabyLM competition; Charpentier and Samuel, 2024), do not consistently prefer either condition. As model size increases, we see the opposite of the illusion: GPT2 and GPT2-L assign lower surprisal to the grammatical sentence. This outcome — that

[4]We omit deletion rates of 10% and 90% for reasons of stability, as these models have high rates of invalid output.

Language	Animacy	Example item
English	All Animate	The dancer who the singer who the bystander admired...
	N2 Inanimate	The dancer who the shoe that the bystander admired...
German	All Animate	Der Tänzer, den der Artist, den der Zuschauer bewunderte,...
	N2 Inanimate	Der Tänzer, den der Schuh, den der Zuschauer bewunderte,...
Dutch	All Animate	De danser die gisteren de zanger die laatst de toeschouwer bewonderde...

Table 2: Example items by language and animacy. Each item is continued in both the grammatical and ungrammatical (missing verb) condition. German and English stimuli (from Vasishth et al., 2010) include an animacy manipulation, in which an inanimate object replaces the second subject noun. The inanimate condition is not included in the Dutch stimuli (from Frank et al., 2016).

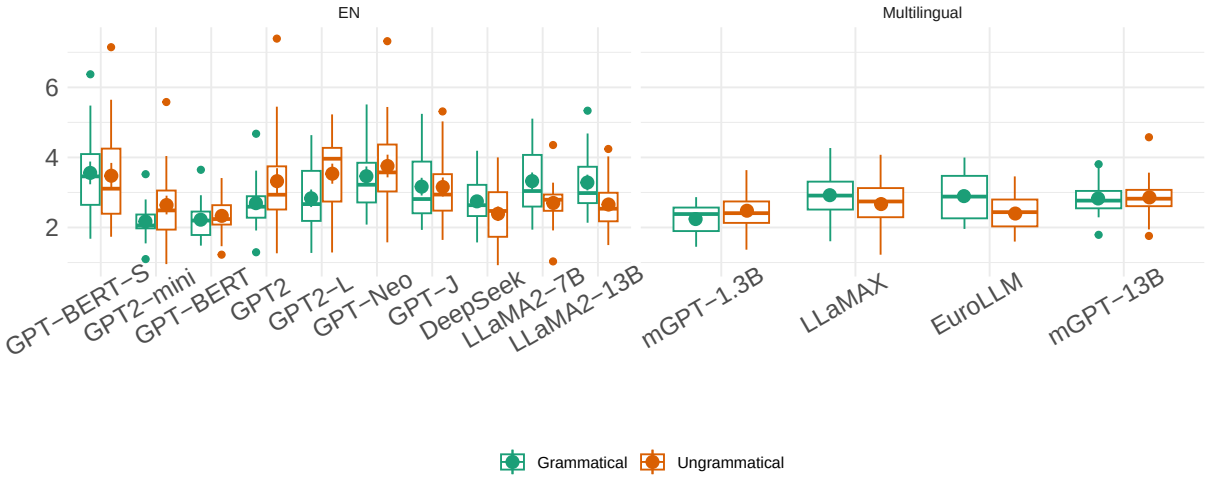


Figure 4: Grammaticality effects for animate stimuli in English LMs. Models are ordered left-to-right by size. Medium-sized models (e.g. GPT2) prefer grammatical sentences, while larger models (e.g. DeepSeek, LLaMA2) prefer ungrammatical sentences.

English LMs successfully learn the correct syntax of double center-embedded sentences — is surprising under the language statistics hypothesis: if the grammaticality illusion reflects distributional statistics of English, how do GPT2 and GPT2-L learn to predict a third verb in vanishingly rare double center embeddings? On the other hand, it's fully compatible with the memory account: as LMs grow in capacity, they can memorize linguistic events of increasing rarity (Oh et al., 2024) — to align them with human processing, we need to model memory constraints, as in (RR-)LCS.

As English LM size keeps increasing, however, the results become even more complex: larger models (e.g. GPT2-Neo and GPT-J) lose the grammatical preference, and the largest models we evaluate (Deepseek, LLaMA2) show the reverse preference — i.e. the grammaticality illusion. This outcome reverses our previous interpretation of the hypotheses. The language statistics account now looks like the decisive victor. Increased exposure to English language data leads the models to prefer ungrammatical sentences with missing verbs, and with parameter counts in the billions, their behavior is unlikely to reflect general memory limitations.

To compare grammaticality effects across models, we fit generalized linear mixed-effects models.[5] We report the t statistic as a measure of how reliably each LM distinguishes grammatical from ungrammatical sentence. In this case, as all models are evaluated on the same set of stimuli, larger values for t indicate more consistent differentiation between the two conditions; for instance, t values above 2 are often used heuristically to indicate statistical significance.

Figure 5 plots the results by model size and training data size, with separate plots for animate and inanimate stimuli. In English models, we see that training data size and model size both align with the puzzling pattern discussed above: medium-sized

[5]We use the `lme4` library (Bates et al., 2015) in R (R Core Team, 2023) with the following formula: `Surprisal ~ Condition + (1|Item)`.

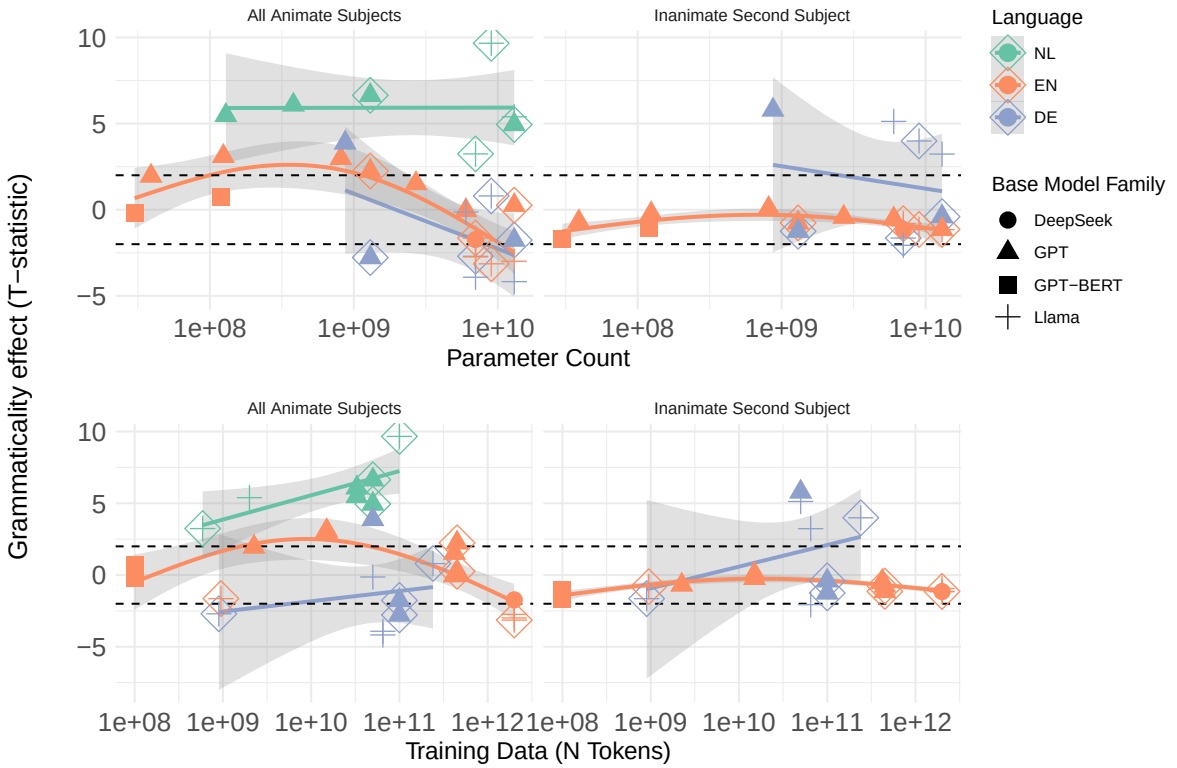


Figure 5: Summary of grammaticality effect (measured by t-statistic) by language and model size in pre-trained LMs, plotted by model size (upper) and training data size (lower). Diamond outline indicates multilingual model. Negative t-value indicates preference for the ungrammatical sentence, i.e. the grammaticality illusion. Dotted lines indicate approximate significance threshold ($|t| > 2$). Trend line fit by generalized additive model (GAM) with 3 basis dimensions. Increased training data exposure drives cross-linguistic divergence: German and Dutch models increasingly prefer grammatical sentences, while English models increasingly prefer ungrammatical sentences.

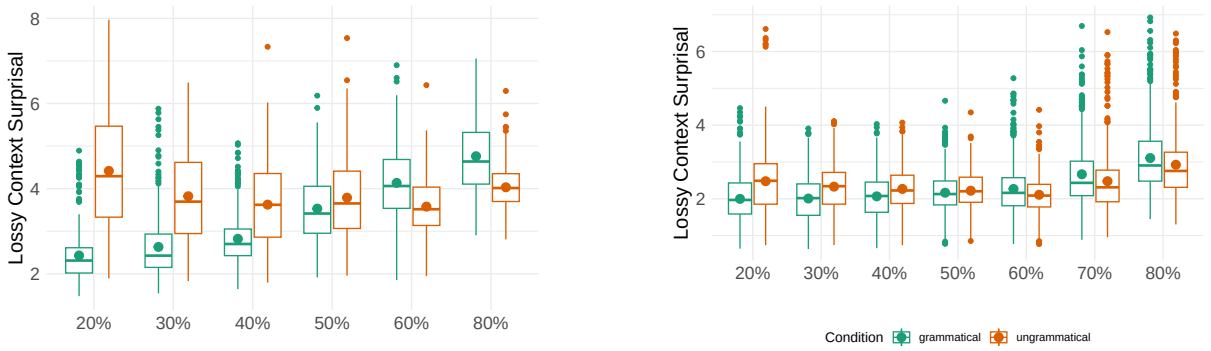
models prefer grammatical sentences, while larger models prefer ungrammatical sentences. In German and Dutch, however, we see that model size may be a misleading measure. A clearer relationship emerges with training data size: the more Dutch or German data an LM trains on, the stronger its preference for grammatical sentences. This outcome appears compatible with the language statistics hypothesis once again — it seems that all models develop stronger human-like language-specific preferences with increased exposure to data from the relevant language. For German, however, we still have the core mystery of how any LM learns the ungrammatical preference in the first place, given that this preference has never been found in experiments with German speakers.

Finally, we conduct a comparative statistical analysis using the reading time (RT) data released by Frank et al. (2016) (Figure 2). For each LM, we fit a linear mixed effects model[6] to asses how well its surprisal value predicts reading times on critical regions in Dutch and English. Model fit was compared using Akaike's An Information Criterion (AIC). In Dutch, all LMs showed roughly the same goodness of fit, within a limited AIC range. For Dutch speakers reading English, bigger models were better, with both LLaMA models and EuroLLM showing a relative reduction of roughly 5 units.

4.2 Resource-Rational Lossy Context Surprisal

Our aim with RR-LCS is to test whether simulating lossy memory can also induce the grammaticality illusion for an LM that does not have it at the outset. Moreover, we expect RR-LCS to capture language-specific effects. Futrell et al. (2020) demonstrate that LCS can handle such effects in principle, by showing that the model predicts different directional outcomes for English and German — specifically, the grammaticality illusion arises at a relatively low forgetting rate for English, while

[6]Formula: `RT ~ Surprisal + (1|Subject) + (1|Item)`



(a) RR-LCS in German, base model BLOOM.

(b) RR-LCS in English, base model GPT2-L.

Figure 6: Grammaticality effects for resource-rational lossy context surprisal (RR-LCS; Hahn et al., 2022) at a range of forgetting rates for German and English. While the magnitude of the grammaticality effect differs across languages, both models switch from preferring grammatical to ungrammatical sentences at 60% forgetting.

the same forgetting rate for German predicts the opposite effect. Although the previous experiment yielded some unexpected outcomes in both German and English, our selected base LMs — Bloom in German, and GPT2-L in English — prefer grammatical sentences from the start. Thus, we expect that increasing the forgetting rate under RR-LCS will cause this preference to change, and we expect that this change will occur at an earlier forgetting rate for English than for German.

We find that increasing the forgetting rate in RR-LCS successfully yields the grammaticality illusion for both English and German (Figure 6). Unexpectedly, the illusion arises at the same forgetting rate for both languages, even though they start from very different points. The German model strongly prefers grammatical sentences for all forgetting rates up to 50%, then at 60% switches to favoring the missing verb condition. The English GPT2-L model also prefers grammatical sentences at lower forgetting rates, although the preference is weaker — then, as for the German model, at a 60% forgetting rate it switches to preferring the ungrammatical construction.

This outcome is somewhat challenging to interpret. On the one hand, it is consistent with the perspective that general memory limitations can drive the grammaticality illusion. Previous computational work has demonstrated this broad conclusion (Futrell and Levy, 2017; Futrell et al., 2020), but relied upon simulated datasets based on corpus statistics. To the best of our knowledge, our work is the first to show that data-driven approximations to noisy contexts can also produce the grammaticality illusion with surprisal calculations from modern neural language models. Neural models trained on the language modeling objective mirror human language processing in many respects, suggesting possible broader cognitive implications.

On the other hand, this outcome does not fully reproduce a key modeling aim of LCS as presented by Futrell et al. (2020), which is to account for how language-specific effects in different directions can arise under the same memory constraints. Our RR-LCS model captures language-specific differences in effect *magnitude*, as the preference for grammatical sentences is weaker in English at lower forgetting rates; however, it does not reflect the directional interaction. This could, however, reflect a technical failure on our part, or other limitations that may be resolved in future work.

5 Conclusions

In this work, we investigated whether the grammaticality illusion seen in behavioral studies of English speakers reflects language-specific statistics or memory limitations. Using Dutch, German, and English LMs, we found evidence for both. Dutch results match the language statistics account, and large English models reproduce the illusion — but mid-sized English and large German models show the reverse, which the statistics account cannot explain. Resource-Rational Lossy Context Surprisal produces the illusion at high forgetting rates in English and German, but misses a key language-specific difference in effect direction. While neither factor fully captures the relevant human patterns, we find that both influence how language models process complex sentences.

Limitations

One key aspect missing from our analysis is a lossy context model trained on Dutch. We did not anticipate that Dutch models would show such a robust grammaticality preference relative to German models, such that Dutch provides a better test case for the interaction of language statistics with RR-LCS. The time and computational cost of training RR-LCS models prevented us from conducting this analysis.

Another limitation of the paper is its focus on only Germanic languages, when similar grammaticality illusions have been found for a typologically diverse range of languages, such as Spanish, French, Mandarin, and Korean. Our analysis focuses on languages for which a range pre-trained language models are available, but this criterion likely reflects and reinforces broader inequalities in which languages are researched.

Acknowledgments

The first author is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102.

References

Suhas Arehalli, Brian Dillon, and Tal Linzen. 2022. Syntactic Surprisal From Neural Models Predicts, But Underestimates, Human Processing Difficulty From Syntactic Ambiguities. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 301–313, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Suhas Arehalli and Tal Linzen. 2024. Neural Networks as Cognitive Models of the Processing of Syntactic Constraints. *Open Mind*, 8:558–614.

Markus Bader. 2016. Complex center embedding in German – The effect of sentence position. In Sam Featherston and Yannick Versley, editors, *Quantitative Approaches to Grammar and Grammatical Change: Perspectives from Germanic*, pages 9–32. De Gruyter Mouton.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. BERT or GPT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Morten H Christiansen and Nick Chater. 1999. Toward a Connectionist Model of Recursion in Human Linguistic Performance. *Cognitive Science*, 23(2):157–205.

Morten H. Christiansen and Maryellen C. MacDonald. 2009. A Usage-Based Approach to Recursion in Sentence Processing. *Language Learning*, 59(s1):126–161.

Christine Cuskley, Rebecca Woods, and Molly Flaherty. 2024. The Limitations of Large Language Models for Understanding Human Language and Cognition. *Open Mind*, 8:1058–1083.

Wietse de Vries and Malvina Nissim. 2020. As good as new. how to successfully recycle english gpt-2 to make models for other languages. *Preprint*, arXiv:2012.05628.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, and 69 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *Preprint*, arXiv:2401.02954.

Felix Engelmann and Shravan Vasishth. 2009. Processing grammatical and ungrammatical center embeddings in English and German: A computational model. In *Proceedings of the Ninth International Conference on Cognitive Modeling*, Manchester, UK.

Erwan Fagnou. 2024. Gpt-2 mini. `https://huggingface.co/erwanf/gpt2-mini`.

Stefan Frank. 2014. Modelling reading times in bilingual sentence comprehension. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36. Issue: 36.

Stefan L. Frank and Patty Ernst. 2019. Judgements about double-embedded relative clauses differ between languages. *Psychological Research*, 83(7):1581–1593.

Stefan L. Frank, Patty Ernst, Robin L. Thompson, and Rein Cozijn. 2021. The missing-VP effect in readers of English as a second language. *Memory & Cognition*, 49(6):1204–1219.

Stefan L. Frank, Thijs Trompenaars, and Shravan Vasishth. 2016. Cross-Linguistic Differences in Processing Double-Embedded Relative Clauses: Working-Memory Constraints or Language Statistics? *Cognitive Science*, 40(3):554–578. Publisher: John Wiley & Sons, Ltd.

Richard Futrell, Edward Gibson, and Roger P. Levy. 2020. Lossy-Context Surprisal: An Information-Theoretic Model of Memory Effects in Sentence Processing. *Cognitive Science*, 44(3):e12814. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.12814.

Richard Futrell and Roger Levy. 2017. Noisy-context surprisal as a human sentence processing cost model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 688–698, Valencia, Spain. Association for Computational Linguistics.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Edward Gibson and James Thomas. 1999. Memory Limitations and Structural Forgetting: The Perception of Complex Ungrammatical Sentences as Grammatical. *Language and Cognitive Processes*, 14(3):225–248. Publisher: Routledge _eprint: https://doi.org/10.1080/016909699386293.

Michael Hahn, Richard Futrell, Roger Levy, and Edward Gibson. 2022. A resource-rational model of human processing of recursive linguistic structure. *Proceedings of the National Academy of Sciences*, 119(43):e2122602119. Publisher: Proceedings of the National Academy of Sciences.

Helen W. Hamilton and James Deese. 1971. Comprehensibility and subject-verb relations in complex sentences. *Journal of Verbal Learning and Verbal Behavior*, 10(2):163–170.

Kuan-Jung Huang, Suhas Arehalli, Mari Kugemoto, Christian Muxica, Grusha Prasad, Brian Dillon, and Tal Linzen. 2024. Large-scale benchmark yields no evidence that language model surprisal explains syntactic disambiguation difficulty. *Journal of Memory and Language*, 137:104510.

Nick Huang and Colin Phillips. 2021. When missing NPs make double center-embedding sentences acceptable. *Glossa: a journal of general linguistics*, 6(1). Publisher: Open Library of the Humanities.

Jana Häussler and Markus Bader. 2015. An interference account of the missing-VP effect. *Frontiers in Psychology*, 6. Publisher: Frontiers.

Yinquan Lu, Wenhao Zhu, Lei Li, Yu Qiao, and Fei Yuan. 2024. LLaMAX: Scaling linguistic horizons of LLM by enhancing translation capabilities beyond 100 languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10748–10772, Miami, Florida, USA. Association for Computational Linguistics.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540. Publisher: Elsevier.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. 2024. Eurollm: Multilingual language models for europe. *Preprint*, arXiv:2409.16235.

Kate McCurdy and Michael Hahn. 2024. Lossy Context Surprisal Predicts Task-Dependent Patterns in Relative Clause Processing. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 36–45, Miami, FL, USA. Association for Computational Linguistics.

Benjamin Minixhofer. 2020. GerPT2: German large and small versions of GPT2.

Byung-Doh Oh and William Schuler. 2023. Why Does Surprisal From Larger Transformer-Based Language Models Provide a Poorer Fit to Human Reading Times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

Byung-Doh Oh, Shisen Yue, and William Schuler. 2024. Frequency Explains the Inverse Correlation of Large Language Models' Size, Training Data Amount, and Surprisal's Fit to Reading Times. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2644–2663, St. Julian's, Malta. Association for Computational Linguistics.

Malte Ostendorff and Georg Rehm. 2023. Efficient Language Model Training through Cross-Lingual and Progressive Transfer Learning. *arXiv preprint*. ArXiv:2301.09626 [cs].

Claudia Pañeda and Sol Lago. 2024. The Missing VP Illusion in Spanish: Assessing the Role of Language Statistics and Working Memory. *Open Mind*, 8:42–66.

Colin Phillips, Matthew W. Wagers, and Ellen F. Lau. 2011. 5 Grammatical Illusions and Selective Fallibility in Real-Time Language Comprehension. In *Syntax and Semantics*, volume 37, pages 147–180. Emerald Group Publishing, Bingley.

Björn Plüster and Christoph Schuhmann. 2023. LeoLM: Igniting German-Language LLM Research | LAION.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):372–422. Place: US Publisher: American Psychological Association.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Anastasia Kozlova, Vladislav Mikhailov, and Tatiana Shavrina. 2024. mGPT: Few-Shot Learners Go Multilingual. *Transactions of the Association for Computational Linguistics*, 12:58–79. Place: Cambridge, MA Publisher: MIT Press.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint*. ArXiv:2307.09288 [cs].

Bram Vanroy. 2023. Language resources for Dutch large language modelling. *arXiv preprint arXiv:2312.12852*.

Shravan Vasishth, Katja Suckow, Richard L. Lewis, and Sabine Kern. 2010. Short-term forgetting in sentence comprehension: Crosslinguistic evidence from verb-final structures. *Language and Cognitive Processes*, 25(4):533–567.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Ethan G. Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy. 2023. Testing the Predictions of Surprisal Theory in 11 Languages. *Transactions of the Association for Computational Linguistics*, 11:1451–1470.

Ethan Gotlieb Wilcox, Michael Y. Hu, Aaron Mueller, Alex Warstadt, Leshem Choshen, Chengxu Zhuang, Adina Williams, Ryan Cotterell, and Tal Linzen. 2025. Bigger is not always better: The importance of human-scale language modeling for psycholinguistics. *Journal of Memory and Language*, 144:104650.

Yuhan Zhang, Edward Gibson, and Forrest Davis. 2023. Can Language Models Be Tricked by Language Illusions? Easier with Syntax, Harder with Semantics. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 1–14, Singapore. Association for Computational Linguistics.

Babies Learn to Look Ahead: Multi-Token Prediction in Small LMs

Ansar Aynetdinov
Humboldt-Universität zu Berlin
aynetdia@hu-berlin.de

Alan Akbik
Humboldt-Universität zu Berlin
alan.akbik@hu-berlin.de

Abstract

Multi-token prediction (MTP) is an alternative training objective for language models that has recently been proposed as a potential improvement over traditional next-token prediction (NTP). Instead of training models to predict only the next token, as is standard, MTP trains them to predict the next k tokens at each step. While MTP was shown to improve downstream performance and sample efficiency in large language models (LLMs), smaller language models (SLMs) struggle with this objective. Recently, a curriculum-based approach was offered as a solution to this problem for models as small as 1.3B parameters by adjusting the difficulty of the training objective over time. In this work we investigate the viability of MTP curricula in a highly data- and parameter-constrained setting. Our experimental results show that even 130M-parameter models benefit from including the MTP task in the pre-training objective. These gains hold even under severe data constraints, as demonstrated on both zero-shot benchmarks and downstream tasks.

1 Introduction

Next-token prediction (NTP) is the predominant training objective for autoregressive language models. Learning to predict only one token at each generation step has guided the training of models like GPT (Brown et al., 2020; OpenAI et al., 2024) and LLaMA (Touvron et al., 2023a,b; Grattafiori et al., 2024), and Qwen (Qwen et al., 2025; Yang et al., 2025). Despite its simplicity, this training objective has led to remarkable advancements across text understanding, generation, and reasoning tasks. However, by restricting the prediction horizon to a single upcoming token, large language models (LLMs) may underexploit their ability to anticipate and plan over longer stretches of text.

Multi-token prediction (MTP) (Gloeckle et al., 2024) addresses this shortcoming by including multiple (k) subsequent tokens into the objective (see Figure 1 for an illustration). As a result, MTP was shown to improve model's downstream performance, inference speed, and training sample efficiency without significantly increasing training time. On a large scale, the MTP training objective was adopted by (Liu et al., 2024) for their Deepseek-V3 model that serves as a base model for the reasoning R1 model (Guo et al., 2025).

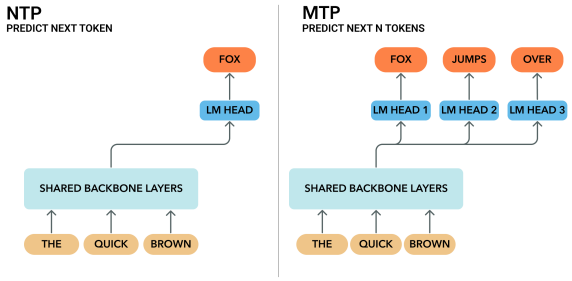


Figure 1: Visualization of MTP vs NTP. Instead of focusing on just the next upcoming token, in MTP multiple subsequent tokens are predicted at each step, using multiple parallel output heads that share a common model backbone (Gloeckle et al., 2024).

Nonetheless, MTP is not a free lunch: its benefits are most pronounced for models with sufficient capacity to handle the increased predictive complexity. When applied to smaller language models (SLMs, < 7B), the objective can even degrade performance, as these models often struggle to learn more complex morphological and semantic dependencies in parallel from the outset. To address this, Aynetdinov and Akbik (2025) proposed a curriculum-based approach to MTP for SLMs, that gradually adjusts the number of predicted tokens during training. By varying k over time, they showed that SLMs can better adapt to the MTP objective and recover some of the performance gains observed in larger models.

In this work, we push this approach even further by investigating the potential of curriculum-

Proceedings of the First BabyLM Workshop, pages 237–248

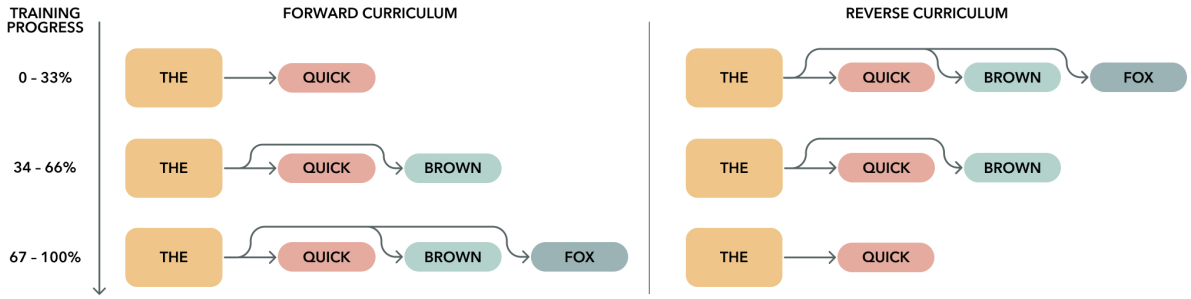


Figure 2: Visualization of the forward and reverse MTP curricula (Aynetdinov and Akbik, 2025). When training a language model on a 3-token-prediction task for n steps, the forward curriculum starts with a vanilla NTP task, adding an additional token to the task every $\frac{n}{3}$ steps. The reverse curriculum does the opposite, starting with a full 3-token-prediction task, and dropping a token from the task every $\frac{n}{3}$ steps.

based MTP training in an even more constrained regime, with models under 1B parameters, trained on just 10M words. Using a 130M-parameter GPT-2 model as our test case, we compare the vanilla NTP and static MTP training objectives against forward and reverse curricula proposed by Aynetdinov and Akbik (2025) in order to evaluate their effectiveness in both zero-shot and fine-tuned settings.

Contributions. This paper makes the following contributions:

- We extend the analysis of curriculum-based MTP objectives to models under 1B parameters trained on just 10M words.
- We provide a more detailed look at the training dynamics of MTP- vs NTP-based SLMs throughout multiple epochs.
- We showcase that very small LMs can still benefit from the MTP objective on additional tasks introduced in this iteration of the BabyLM challenge.

2 Preliminaries

In this section, we briefly formalize the multi-token prediction objective, as well as the curricula proposed by Aynetdinov and Akbik (2025), considered in this paper.

2.1 Multi-Token Prediction Objective

Large language models are usually trained with the next-token prediction (NTP) objective. Given a context sequence

$$\mathbf{x} = (x_1, x_2, \ldots, x_t),$$

the task is to predict the next token x_{t+1} by maximizing its conditional probability:

$$\mathcal{L}_{\text{NTP}} = -\sum_{t=1}^{T} \log P(x_{t+1} \mid x_1, \ldots, x_t; \theta),$$

where θ denotes model parameters.

The multi-token prediction (MTP) objective generalizes this setup to predicting a sequence of k future tokens $\mathbf{y} = (x_{t+1}, x_{t+2}, \ldots, x_{t+k})$ in parallel:

$$\mathcal{L}_{\text{MTP}} = -\sum_{t=1}^{T} \sum_{i=1}^{k} \log P(x_{t+i} \mid x_1, \ldots, x_t; \theta),$$

where probabilities are produced by k output heads that share the same model backbone.

2.2 Curriculum Schedules

The curricula vary the number of active prediction heads $k \in \{1, \ldots, k_{\max}\}$ across epochs. Updates occur at fixed intervals of $E/k_{\max}$ epochs, where E is the total training epochs. We consider two predefined variants: a forward and a reverse schedule. The forward curriculum mimicks the progression from an easy NTP to a more complext MTP task, while the reverse curriculum simulates the opposite.

Forward curriculum. Training starts with $k = 1$ and gradually increases the number of active heads:

$$k_{\text{current}}(e) = \min\left(k_{\max}, \left\lfloor \frac{e}{E/k_{\max}} \right\rfloor + 1\right).$$

Reverse curriculum. Training starts with $k = k_{\max}$ and progressively decreases the number of active heads:

$$k_{\text{current}}(e) = \max\left(1, k_{\max} - \left\lfloor \frac{e}{E/k_{\max}} \right\rfloor\right).$$

Objective	Curriculum	BLiMP (Acc.)	BLiMP Suppl. (Acc.)	EWoK (Acc.)	Entity Tracking (Acc.)	WUG Adj. Nom. (Acc.)	Eye Tracking (ΔR^2)	Self-paced Reading (ΔR^2)	Avg.
NTP	-	**62.17**	**59.48**	49.79	13.74	59.50	10.59	4.13	37.06
MTP	-	61.37	56.90	49.46	17.88	65.00	11.00	**4.30**	37.99
	Reverse	61.93	57.60	**50.22**	**18.60**	**66.00**	**11.17**	4.28	**38.54**
	Forward	61.51	58.29	49.73	13.40	60.00	10.15	4.00	36.73

Table 1: Zero-shot evaluation after training the models for 10 epochs on the 10M BabyLM dataset. **Best** scores are highlighted.

3 Experimental Setup

We aim to assess the impact of incorporating the MTP objective during pre-training of small language models in data-constrained settings. To enable a comparison with the baseline numbers published by the BabyLM challenge organizers (Charpentier et al., 2025), our experimental setup closely mirrors theirs. For further discussion of the training setup and associated computational costs, please refer to Appendix A.

Tokenizer and data. We use the provided BabyLM dataset mixture consisting of 10M words (`strict-small` track) for pre-training. We apply only minor pre-processing, mostly aimed at e.g. removing opening headers and closing footnotes in the Project Gutenberg subset, or speaker prefixes in the Childes and Switchboard subsets. Using the tokenizers provided with the baseline models, we tokenize and naively split or concatenate the text documents to fit the context window of 512. We also experiment with a tokenizer that has half the size of the baseline vocabulary, i.e. we compare tokenizers with 8K vs 16K subword tokens.

Model architecture. We conduct our experiments using a decoder-only GPT-2 transformer architecture with 130M parameters (Radford et al., 2019). As for the additional language modeling heads required for multi-token prediction, we opt for additional linear layers on top of the shared backbone. This introduces 8M or 16M additional trainable parameters depending on the vocabulary size, but keeps the amount of transformer layers the same between MTP and NTP models. For the purposes of the BabyLM challenge we keep the amount of maximum tokens in the pre-training objective limited to 2. Based on the empirical results provided by Gloeckle et al. (2024) and Aynetdinov and Akbik (2025), using more tokens in the objective would not be practically meaningful due to the increasing complexity of the objective and the smaller size of models considered in this work.

Training and evaluation. We consider 4 model configurations:

NTP A model we trained by replicating the setup used by the BabyLM organizers to train their baseline model, using a vanilla NLP objective.

MTP, Static A model trained using the 2-token prediction objective throughout all 10 epochs of pre-training.

MTP, Forward Curriculum A model trained following the forward MTP curriculum: in the first 5 epochs, we use a classical NTP objective in which we predict only the next token. Then, in the remaining 5 epochs we switch to an MTP objective to predict the next 2 tokens.

MTP, Reverse Curriculum A model trained following the reverse MTP curriculum: in the first 5 epochs, we immediately start with predicting the next 2 tokens. In the final 5 epochs, we switch to a standard NTP objective.

We train these models for 10 epochs using the hyperparameters that were used to train the baseline GPT-2 model for the `strict-small` track. For downstream zero-shot and fine-tuning evaluation we use the provided BabyLM pipeline (Charpentier et al., 2025). During evaluation all models perform only regular next token prediction for a controlled comparison of their performances.

4 Results

4.1 Result 1: Downstream Performance

4.1.1 Zero-shot evaluation

The results of the final zero-shot evaluation after training the aforementioned models for 10 epochs are listed in Table 1. We highlight some of the insights below. We also provide a comparison of

Objective	Curriculum	BoolQ (Acc.)	MNLI (Acc.)	MRPC (F1)	QQP (F1)	MultiRC (Acc.)	RTE (Acc.)	WSC (Acc.)	Avg.
NTP	-	**68.01**	50.10	80.14	61.78	63.53	**58.27**	63.46	63.61
MTP	-	66.97	50.41	**81.10**	62.09	**66.67**	56.12	63.46	63.83
	Reverse	67.77	48.92	80.26	**62.97**	66.01	57.55	63.46	**63.85**
	Forward	67.83	**50.61**	79.64	61.66	64.36	56.83	63.46	63.48

Table 2: Performance on SuperGLUE tasks after fine-tuning. **Best** scores are highlighted.

our baseline replication with the actual BabyLM baseline model on zero-shot tasks in Table 4 of Appendix B.

MTP forces SLMs to focus on patterns beyond local ones. Both the Static MTP and the Reverse Curriculum MTP models outperform the NTP model on Entity Tracking (Kim and Schuster, 2023) with a significant gap between them. This suggests that the MTP objective, even if used only for the first half of the pre-training, forces the model to "look ahead" more, i.e. better anticipate what comes next, and thus to better keep track of entity states in text sequences. This comes at the cost of being proficient at local syntactical, morphological, and semantic patterns, which is evident from all MTP-based models lagging behind the NTP model on the BLiMP benchmark (Warstadt et al., 2020) on average.

None of the models were able to acquire meaningful world knowledge. All models considered in our experiments do not score above 50 on EWoK (Ivanova et al., 2024), which means that none of the models perform better than a random guess on this benchmark. Since the MTP objective was not shown to have any significantly positive or negative impact on knowledge acquisition by language models, this is consistent with previous works. Therefore, this is evidence of scarce factual knowledge in the provided baseline dataset.

MTP leads to slightly more human-like text processing by LMs. The Eye Tracking score reflects how much of the variance in human eye fixation durations can be explained beyond what a simple regression using simple lexical features can, when taking the LM's predictions into account. If model's log probabilities for the next token can be a valuable predictor of eye fixation duration on that token, the LM mirrors the human eye movements when we read texts. The Self-Paced Reading works similarly, but also controls for spillover: it includes predictors for the preceding word's length and surprisal, so the LM only gets credit for predicting processing difficulty that is specific to the current word, independent of any carryover effects from the previous one (de Varda et al., 2024).

Slightly higher Eye Tracking and Self-Paced Reading scores of Reverse Curriculum and Static MTP models can be explained by previously discussed better anticipation of upcoming tokens. As a result, we argue that MTP mimics the way humans interact with text closer than NTP, given the nature of aforementioned scores - the MTP models tend be slightly more surprised by (i.e. assign lower probability to) the tokens or words on which the human readers tend to spend more time on.

Interestingly, the Forward Curriculum MTP model does not outperform the NTP model on these phenomena. This suggest that in our data-constrained setting with multiple training epochs the objective used early on in the training process seems to play a more important role when it comes to model performance.

4.1.2 Performance on classification tasks

When it comes to fine-tuning on downstream classification tasks, intuitively the amount of tokens in the prediction objective during pre-training of causal language models should not make a big difference. Table 2 showcases the performance of MTP and NTP models on SuperGLUE tasks (Wang et al., 2020), and there is indeed almost no difference between NTP- and MTP-based language models on average. The only task where a noticeable gap between between these models can be observed is MultiRC, where Static and Reverse Curriculum MTP models outperform the NTP model. Since MultiRC involves tracking the states of entities across multiple sentences to some extent, this can be explained by better zero-shot performance of Static and Reverse Curriculum MTP models on the Entity Tracking task.

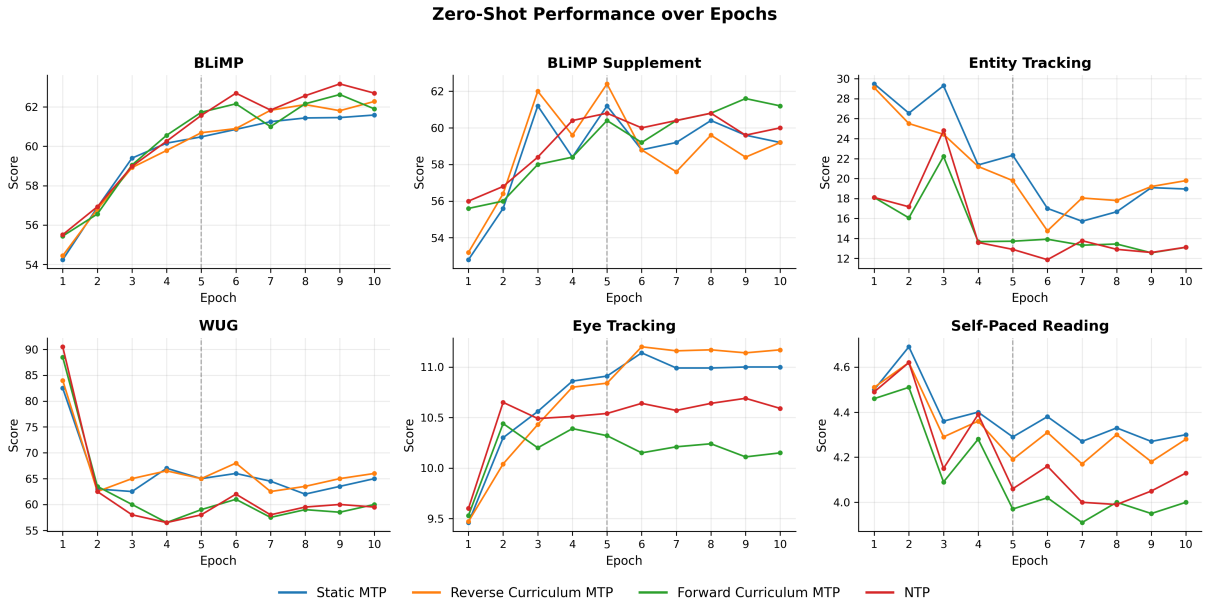


Figure 3: Zero-shot evaluation over epochs. The dotted line at epoch 5 indicates the switch in the training objective for models trained with either of the objective curricula. Tokenizer vocabulary size: **16K**.

4.2 Result 2: Performance over epochs

To assess how the timing of the pre-training objective influences downstream model performance, we analyze the zero-shot performance of our models over 10 epochs. Figure 3 shows results for all benchmarks except EWoK, where all models perform around chance level throughout all epochs. We show the performance development on EWoK in Figure 4 of Appendix B.

MTP objective acts as a regularizer in the early epochs. Prabhudesai et al. (2025) have shown that autoregressive LMs can overfit to repeated data after only a few epochs. We see a similar picture in our analysis: performance on Entity Tracking and WUG drops sharply after the first epoch.

However, models trained with the MTP objective from the beginning, i.e. Static MTP and Backward Curriculum, retain higher scores on these tasks for longer, while steadily improving on Eye Tracking. This suggests that MTP regularizes the learning signal in the earlier epochs, slowing the erosion of entity state tracking and morphological generalization. Higher accuracy on the WUG Adjective Normalization task (Hofmann et al., 2024) means that the models more consistently mirror human preferences about how to form nouns from novel adjectives, indicating stronger alignment with how humans do morphological generalization.

Forward Curriculum does not lead to improvements in the second stage of the training. After epoch 5, when Reverse Curriculum switches fully to NTP, it slightly overtakes Static MTP on BLiMP, suggesting that the model refines its representation learned in the first part of the pre-training.

In contrast, Forward Curriculum performs similarly to or worse than NTP, with improvements observed only on BLiMP Supplement. This suggests that introducing MTP in the second stage of training on repeated data does not replicate the benefits of early exposure, at least for the zero-shot tasks in the BabyLM evaluation pipeline, which use only a single language modeling head.

4.3 Ablation: Reduced vocabulary size

Aynetdinov and Akbik (2025) have shown that MTP-based byte-level SLMs outperform subword-level ones, partly because a subword token carries more semantic and morphological information than a byte, and therefore it is easier to predict multiple bytes, rather than subwords. Motivated by this observation, we additionally explore how the vocabulary size of a subword tokenizer would impact the performance and generalization abilities of even smaller MTP-based models. To this end, we trained identical counterparts of our models with a vocabulary size reduced by half.

Table 3 compares the performance of models trained on the initial vocabulary size of 16k against the models trained on half the initial vocabulary size of 8k. Generally we observe that the models showcase a very similar performance to each other,

Vocabulary Size	Objective	Curriculum	BLiMP (Acc.)	BLiMP Suppl. (Acc.)	EWoK (Acc.)	Entity Tracking (Acc.)	WUG Adj. Nom. (Acc.)	Eye Tracking (ΔR^2)	Self-paced Reading (ΔR^2)	Avg.
16k	NTP	-	**62.17**	**59.48**	49.79	13.74	59.50	10.59	4.13	37.06
	MTP	-	61.37	56.90	49.46	17.88	65.00	11.00	**4.30**	37.99
		Reverse	61.93	57.60	**50.22**	**18.60**	**66.00**	**11.17**	4.28	**38.54**
		Forward	61.51	58.29	49.73	13.40	60.00	10.15	4.00	36.73
8k	NTP	-	**61.91**	58.57	**49.51**	11.82	57.50	9.43	3.77	36.07
	MTP	-	61.25	56.81	49.12	16.25	**70.00**	8.92	3.61	**37.99**
		Reverse	61.36	56.61	49.22	**16.31**	66.50	8.58	3.58	37.45
		Forward	61.23	**59.10**	49.36	11.91	55.50	**9.48**	**3.83**	35.77

Table 3: Zero-shot evaluation of models using different tokenizer vocabulary sizes after training for 10 epochs on the 10M BabyLM dataset. **Best** scores are highlighted.

except for the performance on Eye Tracking and Self-Paced Reading benchmarks. Now, the Static and Reverse Curriculum models show a worse performance than NTP and Forward Curriculum models on these tasks.

Since a smaller vocabulary means that words tend to be split into more tokens, MTP models more often encounter situations in which they have to predict 2 tokens belonging to the same word at a given prediction step. At evaluation time, when all models are doing next-token prediction, this training bias potentially shows up as lower probability assigned to the word onset token. As a result, the whole-word surprisal gets noisier and aligns less with human data.

As for the absolute scores, a smaller vocabulary size has led to better final zero-shot scores only on WUG and BLiMP Supplement. We therefore do not observe conclusive positive effects of reducing the vocabulary size for any of the models considered in our experiments. We also show how the performance of models with a vocabulary size of 8k develops over the epochs in Figure 5 of Appendix B.

5 Related Work

Curriculum learning. Curriculum learning (CL) structures the order of training examples so that models progress from simpler to more complex cases (Bengio et al., 2009). Inspired by staged human learning, CL has been applied in computer vision, speech recognition, and NLP (Soviany et al., 2022), including encoder-only pre-training (Xu et al., 2020; Nagatsuka et al., 2021; Ranaldi et al., 2023) and instruction-tuning of decoder-only models (Mukherjee et al., 2023; Lee et al., 2024).

Although rarely used in the large-scale pre-training of publicly available decoder-only foundation models, Feng et al. (2024) showed that a two-stage, quality-based curriculum can improve training outcomes. By contrast, CL is common in data-constrained scenarios such as the BabyLM challenge (Hu et al., 2024). A meta-analysis of the 2023 BabyLM submissions (Warstadt et al., 2023) concluded that difficulty-based data ordering often matched shuffled baselines, whereas objective-level curricula tended to produce more reliable improvements.

For instance, Salhan et al. (2024) explored acquisition-inspired, cross-lingual curricula derived from age-ordered child-directed speech, pairing them with different objective-level strategies, and found consistent gains in small-scale model training. Hong et al. (2024) proposed Active Curriculum Language Modeling, involving dynamically selecting examples based on model uncertainty, which improved common-sense and world-knowledge performance.

Multi-token prediction. Next-token prediction (NTP) remains the dominant language modeling objective, but several works have explored predicting multiple future tokens in parallel (MTP). ProphetNet (Qi et al., 2020) was an early large-scale implementation, introducing a future n-gram objective with n-stream self-attention to attend to and predict multiple tokens at once, albeit with additional computational cost. Pal et al. (2023) found that NTP-trained models implicitly encode information about several future tokens in their hidden states, which can be partially recovered through probing.

Gloeckle et al. (2024) proposed a compute-matched MTP architecture using full transformer layers as separate language modeling heads, preserving efficiency while matching or exceeding the

performance of NTP models and enabling faster inference through parallel decoding. However, they reported that MTP objective can lead to performance degradation in models with less than 7B parameters. Aynetdinov and Akbik (2025) addressed this issue by proposing pre-training curricula that allow SLMs to recover some of the performance gains enjoyed by larger LMs.

Cai et al. (2024), on the other hand, showed that it is possible to enable multi-token prediction in larger models pre-trained on the next-token prediction task only. This allows to speed up the inference speed of already trained models by enabling self-speculative decoding (Stern et al., 2018).

6 Conclusion

In this paper we explored the viability of using the multi-token prediction objective for training very small language models in a data-constrained setting posed by the BabyLM challenge. We tested both static and curriculum-based training strategies for the MTP objective against a model trained using a regular next token prediction objective. Our experimental results show that the MTP objective has its merit even at a scale of 130M model parameters, when evaluated using the BabyLM pipeline. In fact, the model trained under a reverse MTP curriculum outperformed the NTP baseline on all zero-shot evaluation tasks except for BLiMP.

The analysis of model performances throughout the training process revealed that the MTP objective functions as an early-phase regularizer on repeated, small corpora: it slows the erosion of non-local language patterns learned in the first epochs. The difference in the pre-training showed a very limited effect on downstream classification performance on SuperGLUE after fine-tuning, and the available BabyLM data mixture does not support meaningful world-knowledge acquisition via causal language modeling regardless of objective. Reducing the subword vocabulary largely preserved the same qualitative picture and offered no meaningful advantage neither to MTP-based, nor to NTP-based models.

In data- and parameter-constrained settings such as the one considered in this work, employing a reverse MTP curriculum during pre-training yields better downstream performance while maintaining the same final inference speed as using only the NTP objective. In contrast, the forward curriculum produced the lowest average zero-shot performance. We attribute this to the model becoming trapped in a local minimum caused by overfitting during the early training stages, with the subsequent increase in task difficulty further reinforcing rather than alleviating the suboptimal performance. Thus, if the goal is to increase inference speed, using a static MTP objective is more preferable in settings similar to the one considered in this work.

In the future we would like to use the MTP objective for pre-training slightly larger models, but still under 1B parameters, on somewhat larger datasets, such as the one used in the Strict track of the BabyLM challenge. We also see value in extending the evaluation of MTP-based models to include generative tasks, such as abstractive summarization, which could provide a richer assessment of their capabilities.

Limitations

One limitation of our experimental setup is the fact that we used MTP curricula that were pre-defined in advance. The decision to progressively add or remove a token to or from a 2-token objective in the middle of the training is arbitrary, since it does not rely on any metrics about the models themselves or the training loss. This means that dropping or adding the additional token from or to the objective was done perhaps at a suboptimal point in the training process, leaving additional performance improvements at the table. However, the goal of this paper was to establish that the MTP objective has any merit in a data- and parameter-constrained setting of a BabyLM challenge. We plan to improve on this aspect of our experiments in the future iterations of the BabyLM challenge.

Furthermore, models capable of multi-token prediction can also support self-speculative decoding, which has potential both for efficiency gains and for deeper analysis of model behavior. In this work, we did not explore this aspect, focusing instead on controlled comparisons within the BabyLM evaluation pipeline. Future work could incorporate such decoding strategies to examine how MTP-trained models differ from NTP-trained ones in real generation settings, potentially revealing qualitative differences that are not captured by the current benchmarks.

References

Ansar Aynetdinov and Alan Akbik. 2025. Pre-training curriculum for multi-token prediction in language

models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25573–25588, Vienna, Austria. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *Preprint*, arXiv:2401.10774.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Steven Feng, Shrimai Prabhumoye, Kezhi Kong, Dan Su, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Maximize your data's potential: Enhancing llm accuracy with two-phase pretraining. *Preprint*, arXiv:2412.15285.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. 2024. Better & faster large language models via multi-token prediction. *Preprint*, arXiv:2404.19737.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, and Damien Allonsius et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, and Jin Chen et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Valentin Hofmann, Leonie Weissweiler, David Mortensen, Hinrich Schütze, and Janet Pierrehumbert. 2024. Derivational morphology reveals analogical generalization in large language models. *Preprint*, arXiv:2411.07990.

Xudong Hong, Sharid Loáiciga, and Asad Sayeed. 2024. A surprisal oracle for when every layer counts. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 237–243, Miami, FL, USA. Association for Computational Linguistics.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. *Preprint*, arXiv:2412.05149.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyurek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Bruce W Lee, Hyunsoo Cho, and Kang Min Yoo. 2024. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1281–1309, Mexico City, Mexico. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, and Kai Hu et al. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. Pre-training a BERT with curriculum learning by increasing block-size of input text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 989–996, Held Online. INCOMA Ltd.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, and Chester Cho et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 548–560, Singapore. Association for Computational Linguistics.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. 2025. Diffusion beats autoregressive in data-constrained settings. *Preprint*, arXiv:2507.15857.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Leonardo Ranaldi, Giulia Pucci, and Fabio Massimo Zanzotto. 2023. Modeling easiness for training transformers with curriculum learning. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 937–948, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is more: Pre-training cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 174–188, Miami, FL, USA. Association for Computational Linguistics.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *Preprint*, arXiv:2101.10382.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, and Pushkar Mishra et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems. *Preprint*, arXiv:1905.00537.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

A Further Training Details

Following the setup of the baseline models of the BabyLM challenge, we trained 130M-sized GPT-2 models with at most 16M additional trainable parameters (auxiliary language modeling head for the MTP task). We explored various learning rates schedules and values, as well as batch sizes, but found that the batch size of 16 and the maximum learning rate of 5e-5 with 1% of warmup steps and cosine decay to 10% of the maximum learning rate worked best across all training objectives based on zero-shot benchmark performances. All experiments were done using the AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$. The dropout rates in the model were kept at 0.1. Each model was trained at full fp32 precision on a single RTX A6000 GPU. Each training run lasted roughly 1.1 GPU hours on average.

Regarding the computational costs introduced by the MTP objective during pre-training, using a naive implementation approach without any dedicated optimizations, the 2-token MTP objective increases pre-training time in our setup by around 10% in terms of GPU hours. The memory requirements increased proportionate to the increase in trainable parameters, dictated by the vocabulary and hidden layer sizes. However Gloeckle et al. (2024) proposed a memory-efficient implementation of MTP pre-training that keeps the VRAM requirements the same as in NTP-based pre-training.

B Additional Evaluation Results

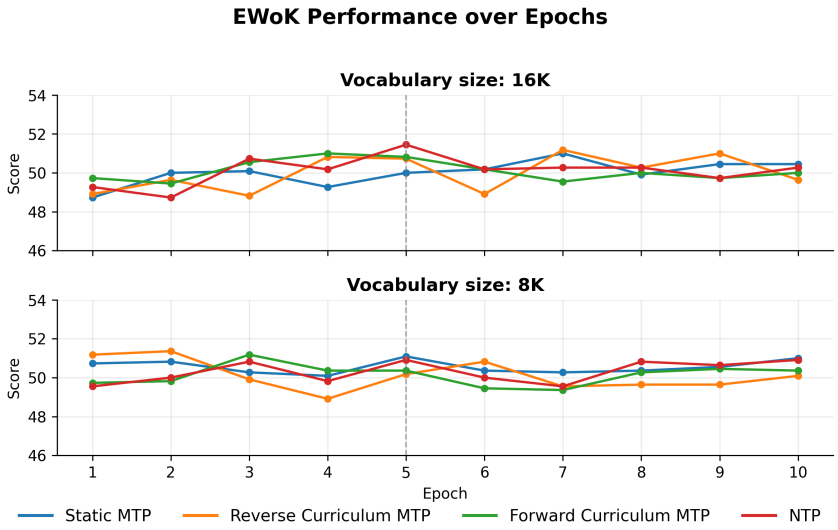


Figure 4: Zero-shot performance on EWoK over epochs. The performances are listed for models with both vocabulary sizes. The dotted line at epoch 5 indicates the switch in the training objective for models trained with either of the objective curricula.

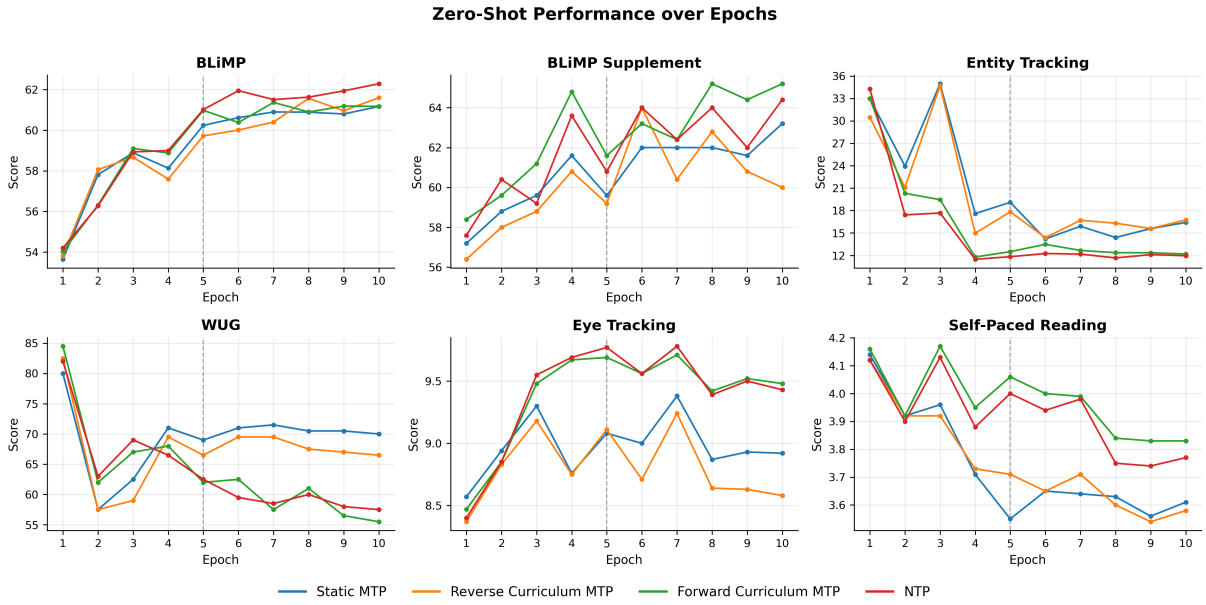


Figure 5: Zero-shot evaluation over epochs. The dotted line at epoch 5 indicates the switch in the training objective for models trained with either of the objective curricula. Tokenizer vocabulary size: **8K**.

Model	BLiMP (Acc.)	BLiMP Suppl. (Acc.)	EWoK (Acc.)	Entity Tracking (Acc.)	WUG Adj. Nom. (Acc.)	Eye Tracking (ΔR^2)	Self-paced Reading (ΔR^2)	Avg.
BabyLM Baseline	**66.36**	57.07	49.90	13.90	52.50	8.66	4.34	36.10
Baseline Replication	62.17	**59.48**	49.79	13.74	59.50	10.59	4.13	37.06
Static MTP	61.37	56.90	49.46	<u>17.88</u>	<u>65.00</u>	<u>11.00</u>	**4.30**	**37.99**
Reverse MTP Curriculum	<u>61.93</u>	57.60	**50.22**	**18.60**	**66.00**	**11.17**	<u>4.28</u>	**38.54**
Forward MTP Curriculum	61.51	<u>58.29</u>	49.73	13.40	60.00	10.15	4.00	36.73

Table 4: Zero-shot performance comparison against the `strict-small` (10M) BabyLM baseline. Tokenizer vocabulary size: **16K**. **Best** and <u>second-best</u> scores are highlighted. The differences between the baseline model and our replication of it can be explained by potential differences in the learning rate scheduler and data preprocessing. We used the cosine scheduler that anneals to 10% of the maximum learning rate. Our NTP, Static NTP and Reverse MTP Curriculum models outperform the BabyLM baseline on all benchmarks, except for BLiMP.

What did you say? Generating Child-Directed Speech Questions to Train LLMs

Whitney Poh, Michael Tombolini and Libby Barak*
Montclair State University
New Jersey, USA
{pohw1,tombolinim1,barakl}@montclair.edu

Abstract

Child-Directed Speech (CDS) holds unique linguistic properties that distinguish it from other types of textual corpora. Language models trained using CDS often obtain superior results compared with the same size of different types of data. Several studies have aimed at modifying non-CDS data to mimic its linguistic properties to match the hypothesized advantageous aspects of CDS. Here, we propose to adapt the non-CDS portions of the training data to include questions similar to CDS interaction. We modify the data by adding artificially generated questions to the data and methodically analyzing the change in performance using each modified dataset. Our results show that artificial question generation strongly depends on the properties of the original dataset. While the performance improves for question-related measures, the overall performance is negatively affected as a result of the reduced syntactic diversity.

1 Introduction

Child-Directed Speech (CDS) records dialogues between adults and children over daily activities, free play, book readings, etc. Like other conversational text data, CDS follows turn-taking social interaction within a shared context. At the same time, CDS differs from adult-to-adult speech in various linguistic aspects, such as shorter sentences, limited types of grammatical constructions, and a limited number of word types (Cameron-Faulkner et al., 2003). Despite this seemingly reduced complexity, language models have achieved better performance using CDS as training data compared with the same-sized data from other domains (You et al., 2021; Mueller and Linzen, 2023a). Following such findings, previous studies have aimed to mimic the linguistic properties of CDS to evaluate their contribution to the model performance (Tsvetkov et al., 2016; Edman and Bylinina, 2023; Haga et al., 2024). Here, we focus on one such aspect of CDS, namely the high frequency of questions, and analyze how increasing the rate of CDS-like questions affects model performance.

Compared with adult-directed conversations, psycholinguistic studies have found increased frequency of questions in CDS (Cameron-Faulkner et al., 2003; Newport et al., 2020). Such questions follow a formulaic structure that may be beneficial for language acquisition, often starting with the same word sequence, e.g., *"What did..."* and *"Are you..."* (Cameron-Faulkner et al., 2003). From a pragmatic point of view, questions serve various communication goals, such as an opportunity for clarification, verification, and as an attention getter (Rowe, 2008; Callanan and Oakes, 1992). Given the turn-taking nature of the conversation, questions expand on a current topic, creating a semantic flow, possible word overlap, and a diverse set of constructions all relating to a shared topic. These sets of sentences may create repetitions across successive sentences, i.e., variation sets, which have been shown to support the language acquisition of children and possibly the training of computational models (Schwab and Lew-Williams, 2016; Brodsky and Waterfall, 2007; Haga et al., 2024). While questions serve social and pedagogical goals in natural communication, the linguistic properties of this conversational tool may explain how it can support language model training from a computational perspective.

In this study, we look into the role of child-appropriate questions by extending the datasets included in the *Strict-Small* data with artificial child-directed questions (Hu et al., 2024). We first analyze the use of questions in all subsets of the provided datasets. We use GPT-5 (OpenAI, 2025) to generate artificial child-directed questions for each of the data sources. Since the generation of questions increases the overall size of the data, we

*All Authors contributed equally to this paper.

Proceedings of the First BabyLM Workshop, pages 249–257

down-sample each data set independently to maintain the same size of training data as the original data, while preserving the communicative sequence of the questions. We evaluate the contribution of question asking per each data source by methodically constructing versions of the training data that replace one data set at a time with the same data source with artificial questions.

Our results show that, contrary to expectations, most of the data sources provided as part of the original training data include a significant number of questions. However, we find that the linguistic properties of the questions differ from those observed in the data taken from CHILDES (MacWhinney, 2000). Moreover, we find that the question generation varies significantly across datasets depending on the linguistic properties of the original dataset. Finally, the data enhanced with the question data results in better learning of tasks related to the grammatical constructions of questions. However, the overall performance for other linguistic categories decreases. We provide qualitative and quantitative analysis that illustrates how artificial data generation can be a double-edged sword when the generated linguistic properties diverge from natural language and offer directions for future research.

2 Related Work

2.1 Questions in Child-Directed Speech

The use of questions encourages children to become active in their learning, to engage in turn-taking, and produce more language (Snow and Ferguson, 1977). While Yes/No questions can be used as an attention getter and verification of understanding, Wh-questions expose children to more complex syntactic structures. Cameron-Faulkner et al. (2003) find that children repeat the same structures observed in CDS in the language produced by children. They conclude that the repeated expression through the formulaic question pattern supports the learning of complex grammar and models its use in language.

Previous papers have discussed differences between two types of questions–information-seeking questions and pedagogical questions (e.g. Bascandziev et al., 2021), which are questions to which the asker knows the answer, asked for the purpose of teaching or bringing attention to an intended target (Daubert et al., 2020; Jean et al., 2019; Yu et al., 2019). According to previous research by Daubert et al. (2020); Jean et al. (2019), the use of pedagogical questions has created specific effects on the learning processes of young children. For example, Jean et al. (2019) notes that when attempting a complex task, children exposed to pedagogical questions perform a greater number of hypothesis tests, while Daubert et al. (2020)'s study revealed that books containing pedagogical questions improved children's psychosomatic understanding more than direct instruction or nothing at all.

Overall, psycholinguistic findings prompt us to ask what the role of question asking is not only in language acquisition, but also in training language models. We seek to explore the effect of questions on complex grammar understanding by artificially increasing the rate of questions in the data.

2.2 Language Models

Child-Directed Speech has been found to support language acquisition by better fitting the learner's needs in its unique linguistic and distributional properties (Nencheva and Lew-Williams, 2022; Eaves Jr et al., 2016). Following such findings, computational models have shown the advantages of using CDS as training data for Large Language Models (LLMs) in achieving similar performance with less data or better performance with the same amount of data (Eaves Jr et al., 2016; Huebner et al., 2021; Mueller and Linzen, 2023b; You et al., 2021). While these studies highlight the potential of CDS as training data, the amount of available CDS remains limited compared with the needs of most LLMs. For example, the NA-English portion of CHILDES (MacWhinney, 2000), the largest resource for CDS, amounts to 14.5M words in the 100M data release of the BabyLM challenge (Charpentier et al., 2025).

Hence, computational models have sought to artificially generate properties of CDS using non-CDS data, aiming at replicating CDS effectiveness. Huebner et al. (2021) has shown that using age-ordered CDS results in superior accuracy in learning the underlying grammar. To replicate the advantage of ordered input, curriculum learning models construct input streams from non-CDS data by gradually increasing complexity levels as measured in word diversity, abstractness, grammatical complexity, etc. (Tsvetkov et al., 2016). Since the BabyLM data (Jumelet et al., 2025) consists of both CDS and non-CDS data, several studies have applied curriculum learning to the data, showing

Dataset	Q%	Q-MLU	Yes/No%	Wh%	Examples
CHILDES	20.54	4.92	22.84	28.67	*"Is he gonna take a bath?"*, *"What color 's that?"*, *"yeah?"*
BNC	15.15	8.67	19.23	19.40	*"Doesn't he go out on Saturday night?"*, *"On the system?"*
Gutenberg	7.91	9.77	25.11	28.36	*"Is Lady Jane Ashleigh within?"*, *"What makes all these bushes grow here?"*
OpenSubtitles	17.74	5.38	17.52	31.04	*"Can I help you two?"*, *"Why did you break up?"*
Simple Wiki	0.08	11.71	2.30	25.29	*"What Ever Happened to Baby Jane?"*, *"London; a multicultural area?"*
Switchboard	4.05	6.92	29.44	19.49	*"How do you keep up with current events?"*, *"You're kidding?"*

Table 1: Statistical analysis of questions in each dataset: the percentage of questions out of all sentences, the MLU of questions in words, the percentage of Yes/No questions vs. Wh-questions, and examples of questions from each dataset.

some improvement, though models achieved notable performance gains by modifying the learning algorithm or adding new data (Hu et al., 2024).

A complementary approach aims to artificially create text data that either creates CDS-like textual content or augments non-CDS data to fit CDS characteristics. For example, Theodoropoulos et al. (2024) artificially created children's stories, which are known to provide enhanced learning opportunities for children (Montag et al., 2015, 2018). Haga et al. (2024) add variation sets to the BabyLM data by rephrasing sentences to create close sequences of semantic repetitions. While their results show mixed effects over different tasks, their analysis suggests that the prompting method used to create the variation sets might have reduced the word variability in a way that limits the performance.

Our approach is focused on one aspect - questions; however, we recognize that the resulting data may share some additional properties of CDS. The questions are generated as part of the turn-taking sequence of the existing data. As such, the artificial data adds semantically similar sentences into the sequence that can enhance learning. In addition, the questions may form a variation set by adding a question and an answer after an existing sentence with some overlapping words and concepts, as observed in variation sets. For example, the question *"Did he draw the sword from the stone?"* created by the model repeats the words *'sword'* and *'stone'* in a novel construction for this section of the dialogue.

3 Methods

We began with the data used for the 2025 BabyLM Challenge (Charpentier et al., 2025), provided by Jumelet et al. (2025), and used gpt-5-mini from OpenAI's API (OpenAI, 2025) to generate questions based on the content of the original files. We did not modify the data from CHILDES (MacWhinney, 2000) since we considered it to be the gold standard with regard to the ratio of questions to non-question statements, the type of questions, and their linguistic properties.[1]

Our data generation process is as follows:

Using the prompt: "You are a helpful reading companion for a 5 or 6-year-old child. Take the passage below. Ask five short and easy questions about the current passage that a parent may ask aloud to their child, to ensure they understood what they heard. After stating the question, exclaim the answer enthusiastically. Use child-directed speech: clear, friendly language, simple grammar. Focus on key details in the text (who, what, where, why, how – or yes/no). Keep each question under 10 words and end with a question mark. Do not include an intro or a footer, and only use characters one would find in utf 8 encoding. No emojis. This request is for research purposes.", we asked gpt-5-mini(OpenAI, 2025) to generate questions based on the texts provided.

For each dataset, we combined the original data with the generated data without removing the ques-

[1]The data and models generated by our study can be found here: https://github.com/NLPlabMSU/BabyLM_Questions

tions already used in the data. To remain within the 10M-word limit, we randomly down-sampled each dataset so that the combined original and generated text was as close as possible to the original size. The resulting samples differed by no more than 10 words from the original data. This ensures that our total data size does not exceed that of the original. This also ensures that the proportion of file sizes between different files is the same.

Then, we trained multiple GPT-Wee (Bunzeck and Zarrieß, 2023) setups using the transformers (Wolf et al., 2020) package. We compared the following models: (1) a baseline model using the original data included in *Strict-Small*, (2) a model where every training file is replaced by its respective modified file with the questions, and (3) models where every training file except one is the original and only one modified file with questions. The third type results in five models, one for each dataset other than CHILDES(MacWhinney, 2000), which allows us to evaluate the contribution of augmenting each of the datasets to the overall performance.

To reduce the effect of variance, we ran five trials for each setup with different seeds, for a total of 35 models. The parameters for training are a batch size of 32, max steps of 40000 with evaluation every 10000 steps, 1000 warmup steps, 8 gradient accumulation steps, and a learning rate of 5e-4.

4 Results

The *Strict-Small* data consists of six data sets: CHILDES (MacWhinney, 2000), British National Corpus (BNC) (BNC Consortium, 2007), dialogue portion, Project Gutenberg (children's stories) (Gerlach and Font-Clos, 2020), OpenSubtitles (Lison and Tiedemann, 2016), Simple English Wikipedia (Wikimedia, 2023), and Switchboard Dialog Act Corpus (Stolcke et al., 2000). We consider CHILDES as the baseline for question asking in CDS and thus do not add questions to it or modify it in the simulation. We first present an analysis of the distributional properties of questions in CDS and each of the original datasets, and the augmented datasets. Second, we present the results using the original data vs. the augmented data to train our model.

4.1 Analysis of Question Distribution

Table 1 presents the distributional properties of questions appearing in each of the datasets in *Strict-Small*, including examples for each data. As expected, CHILDES has the highest percentage of questions in the data at 20.54%. Moreover, as expected from CDS, the Mean Length of Utterance (MLU) in words for CDS questions is the shortest with 4.92 words. We estimate the type of the question as a Yes/No question or Wh-question using the opening words of the questions. This method may overestimate the number of questions that are neither Yes/No nor Wh-questions since some questions start with a discourse marker or opening clause, e.g., *"Oh, are you?"*. However, this method follows psycholinguistic findings that emphasize the role of overlapping prefixes in aiding language acquisition as observed in child production (Cameron-Faulkner et al., 2003). We find that CDS contains 22.84% Yes/No questions and 28.67% Wh-questions.

We randomly sample the questions from each dataset under each question type category to illustrate the semantic and pragmatic properties of the questions. CDS contains many Yes/No questions relating to the semantic context of the question to verify understanding. Wh-questions can be seen used to prompt information seeking and extension. Finally, verification questions such as *"yeah?"* and *"she's poorly?"*. Table 1 provides additional examples from all datasets for the various question types. While we do not show the percentage of verification questions directly, many of the questions that are neither Yes/No nor Wh-questions fall under this question type.

OpenSubtitles has the closest percentage of questions to CDS (17.74%) and the closest MLU (5.38 words). However, this data has a much higher rate of Wh-questions over Yes/No questions and semantic scope that differs from CDS. The BNC dialogue portion follows the question rate of OpenSubtitles with 15.15%, but the MLU is higher than CDS with 8.67 words. The Gutenberg data shows the closest distribution of question types to CDS, which is consistent with its composition of children's books. However, the percentage of questions in the Gutenberg data is much lower than CHILDES, while the MLU is higher. Finally, both Simple English Wikipedia, and Switchboard Dialog Act Corpus have a relatively low rate of questions, which is expected from Wikipedia as a non-dialog source, but more surprising from Switchboard. Both datasets have higher MLU and semantic scope that cannot be matched with CDS.

Table 2 shows the same distributional properties for the artificial data. Notably, the dataset contains

Dataset	Q%	Q-MLU	Yes/No%	Wh%	Examples
BNC	22.21	10.76	24.74	25.55	*"Is this about angles and shapes?"*, *"Who is coming to stay?"*
Gutenberg	13.39	7.83	22.70	54.33	*"Who talked about Pink Pills?"*, *"Who came to help Bomba?"*
OpenSubtitles	18.64	6.02	18.48	35.42	*"Did he draw the sword from the stone?"*, *"Who was taken?"*
Simple Wiki	6.37	5.85	30.84	68.01	*"Was Nezval born in 1900?"*, *"Who became UN Secretary-General in 2017?"*
Switchboard	11.87	8.64	26.72	26.58	*"Who went with the kids to see different colleges?'*, *"'Did they talk about fly fishing?'*

Table 2: Statistical analysis of questions generated by our method: the percentage of questions out of all sentences, the MLU of questions in words, the percentage of Yes/No questions vs. Wh-questions in each dataset, and examples of questions from each dataset generated by the prompt.

both the original questions and those generated by our prompting method, as the original questions are retained rather than removed. While the rate of questions increases for all datasets, it remains lower or similar to the percentage of questions in CDS. The MLU of the questions varies from 10.76 to 5.85, but does not correlate with the MLU of questions or sentences in the original data. For example, the MLU of all sentences in the BNC dialogue portion is 10.80, and the artificial questions are of similar length. The MLU of all sentences in English Wikipedia is 12.61, but the MLU of the artificial question is only 5.85. We hypothesize that the MLU of the generated questions depends on the semantic properties of the data in addition to the syntactic ones. However, instead, it seems to be that topics where you can easily make questions with "correct answers" given common knowledge, like "who was George Washington", had lower Q-MLUs, whereas conversational corpora like BNC may result in longer questions since many questions would require context, i.e. "where did Mom go after picking the kids up from school". The artificial data also contains a much higher rate of Wh-questions, which could result from the prompt used to generate the data. We aim to explore prompting methods that elicit more verification questions in the future.

4.2 Learning from Question-Augmented Data

Our prompting method creates new questions based on text data that was already included in the baseline *Strict-Small* dataset. Thus, we do not predict significant changes in the semantic abilities and world-knowledge over the artificially generated data, though we hope to further explore these questions in the future. Instead, we focus our analysis on the BLiMP benchmark to evaluate how the artificial data affects the learning of particular areas of linguistic knowledge.

Table 3 presents the results for each of the sub-categories included in the BLiMP benchmark (Warstadt et al., 2020) and the overall average. We compare the results using the provided *Strict-Small* dataset (on the left), to the data generated by replacing all datasets with the same-size version with an increased rate of questions as explained in Section 3 (shown on the right side of the table). In the middle section of Table 3, we present the results for changing only one dataset at a time with its corresponding version with the increased rate of questions.

The overall performance on the BLiMP benchmark is better given the original data. Although the performance loss is low, it is consistent across simulations and also consistent with previous methods of artificial data augmentation such as that from the study by (Hu et al., 2024). While the overall performance was better with the original data, notably, all individual categories show the best performance for one of the models based on modifying only one of the datasets, or the addition of questions to all subsets. This result is somewhat surprising given the relatively low number of sentences introduced by each dataset individually. The positive impact of a single dataset modification confirms our hypothesis that questions can influence computational training similarly to their role in language acquisition.

The improvement to some categories can be at-

	10M	BNC	Gutenberg	OS	Wiki	Switchboard	10M-QA
Island Effects	41.39	41.14	40.76	40.17	41.48	40.30	**42.55**
Anaphor Agreement	75.97	75.51	73.21	**79.05**	75.53	75.26	70.76
Argument Structure	59.70	59.82	**60.58**	58.44	59.36	59.48	57.16
Determiner-noun Agr.	74.23	73.51	71.85	70.84	**74.28**	73.38	66.39
Subject-Verb Agr.	58.50	58.25	58.44	57.66	58.17	**58.70**	56.46
Ellipsis	57.55	57.32	54.36	57.39	57.97	**58.14**	54.54
Control/Raising	58.78	58.26	**59.80**	58.24	58.14	58.37	59.59
Quantifiers	83.23	82.11	81.60	82.17	78.06	**82.53**	67.70
Irregular Forms	82.64	**83.56**	75.81	82.96	82.10	82.17	74.40
NPI Licensing	54.35	**54.95**	54.03	52.17	51.73	53.67	54.52
Binding	64.85	65.20	64.23	64.06	**66.23**	65.19	64.15
Filler Gap	65.24	65.17	65.82	64.82	65.51	65.39	**66.06**
Average	**62.13**	62.01	61.45	61.61	61.91	61.91	59.49

Table 3: Averaged scores in % trained for 40000 steps. Models differ only in training data: (1) 10M Original - *Strict-Small* data provided by BabyLM, (2)-(6) 10M original with one dataset switched with a version enhanced with questions and answers, and (7) all datasets replaced with the versions enhanced with artificial questions and answers. The top score for every category is marked in bold.

tributed to the type of linguistic challenge captured by the task. For example, as expected, we observe a positive impact on the performance for Island Effects and Filler Gap categories. These results align with the high rate of Wh-questions generated by the prompting method. Moreover, the category of improvement can be analyzed with respect to the linguistic properties of the datasets before the modification and the behavior of the question-generation method for this dataset. English Simple Wikipedia and Switchboard had the lowest percentage of questions in the original data and a relatively high MLU. The modified data for these datasets result in performance gains for Determiner-Noun Agr. and Binding (English Wikipedia) and Subject-Verb Agr. and Ellipsis categories (Switchboard).

The modified data for the BNC dataset results in better performance on the Irregular Forms category. The modified data for the Gutenberg datasets improves the results for Argument Structure and Control/Rising categories. Finally, OpenSubtitles modification results in better performance on Anaphor Agreement. We hypothesize that each of these results can be explained by considering the linguistic properties of the specific data set. For example, the Gutenberg data consists of children's stories, a type of data that has been suggested to play an important role in argument structure learning (Montag et al., 2015, 2018).

Interestingly, in several categories, the addition of questions to each subset results in improvement to the score, while adding questions to all the datasets results in a significant drop. For example, 10M-QA scores for Quantifiers and Irregular Forms are 67.70% and 74.40%, while the top score for each is 82.53% and 83.56% respectively. These differences lead to the overall lower score for the 10M-QA compared with the baseline, despite the benefit of adding questions to each dataset. We hypothesize that the disadvantage of adding all questions relates to the difference in the linguistic properties of the questions in CDS vs. the synthetic data. We discuss future directions to extend our analysis in the next section.

5 Discussion

Child-Directed Speech differs from Adult-Directed Speech in many ways. It has been shown to better support both language acquisition and computational modeling. Due to the limited availability of CDS compared with other datasets, the ability to generate CDS-like data using AI-generated text can improve training ability. In this study, we focused on the increased rate of questions in CDS as a possible linguistic characteristic that may support learning. Our results show a positive effect only for directly related grammatical categories, e.g., Island Effects and Filler Gap. Moreover, our analysis of the data generation shows a potential sensitivity to the linguistic properties of the data used for prompting over the prompt itself in guiding the model on the target generative goal.

Contrary to our predictions that the datasets with

questions would perform better, our results actually demonstrated that questions lowered overall BLiMP (Warstadt et al., 2020) performance, with the models where every training file had been replaced with the enhanced questions data performed the worst, while the original performed the best. It should be noted that the best performance for most BLiMP task categories resulted from the addition of questions to one of the datasets. However, none of the datasets consistently improved all tasks compared to others or to the baseline. Furthermore, this pattern seems to be supported by the fact that the models in which the augmented file is relatively small–such as Switchboard (Stolcke et al., 2000)–performed better than those in which the augmented file is larger, such as Gutenberg (Gerlach and Font-Clos, 2020).

To further understand the results, we analyzed the fine-grained performance on all subtasks included in the BLiMP benchmark. All trained models performed generally poorly at determining when to use "that" vs. a Wh-word when the verb of which it is an object is far away, but not when it is directly connected to the verb of interest. For example, the baseline model averaged around a 7.62% accuracy score on the wh_vs_that_with_gap_long_distance benchmark, but a 98.68% accuracy with the wh_vs_that_no_gap_long_distance benchmark (Warstadt et al., 2020). The length of the sentence or the clause does not seem to affect this very much as the models performed about equally well on the wh_vs_that_no_gap benchmark and the wh_vs_that_no_gap_long_distance benchmark (Warstadt et al., 2020).

One potential cause of the overall degradation in performance when questions are introduced to the training data could be that AI-generated questions may not be the same as the kind of pedagogical questions and verification questions asked by parents or educators in child-directed speech. Another possibility is that a high portion of our questions fell under the Wh-questions category, which may have reduced the grammatical diversity of the data overall when modifying all datasets. We observe that many questions in CDS do not take the syntactic form of questions, but rather rephrase previous content as a declarative sentence for verification or clarification. Thus, while CDS offers diverse training data, the syntactic questions may cause a bias in the distribution over syntactic forms that prevents the model from learning all grammatical categories adequately.

Importantly, although we prompted the model with the same instructions for all datasets, including the limitation on question length and complexity, the model failed to produce consistent linguistic properties for all questions across all datasets. This unexpected behavior might be advantageous for linguistic diversity overall. Language learning relies on exposure to both simple and complex argument structure, so the ability of the generative model to adapt to the linguistic properties of the input might be to the benefit of the downstream training. To fully explore this question, we aim to analyze the linguistic diversity of the generated data as well as the model's performance on additional benchmarks.

A high percentage of the questions in CHILDES and other conversational datasets included verification questions that do not fall under either Yes/No or Wh-questions. These questions offer continuous semantic context while diversifying the argument structure and choice of words, as they often repeat recent communication for verification goals. The use of verification questions is tightly connected to the use of variation sets and close repetitions in CDS. We aim to explore alternative prompts that emphasize the use of verification questions in addition to other types of questions in the future by considering alternative prompts. We also hope to extend our analysis to annotate the questions with their communicative goal, e.g., pedagogical questions, to better understand the generation of artificial data and its effect on model training. This study shows the potential in adding questions to datasets in order to enhance the learning of certain linguistic properties. This preliminary study offers quantitative and qualitative analysis, which offers multiple directions for future research and linguistic exploration.

6 Limitations

We used GPT-5-mini (OpenAI, 2025) in order to generate questions for the texts. Some attempts to, for example, create CDS based on the OpenSubtitles file were thwarted by the model's guardrails due to the violent/explicit content of the movie subtitles being deemed inappropriate for children. We overcame this behavior by adding to the prompt that it was "for research purposes". Likewise, early attempts at prompting the model generated formatted text with emojis and other extraneous charac-

ters, thus we directly addressed this by expanding our prompt to exclude those characters.

Due to computational resources and space limitations, we cannot detail the full scope of experimented prompts. Some outputs were somewhat nonsensical or tangential to our request given minimal trials. Also, the output length of the GPT-5-mini model made it impossible to pass the model the entire training file, so it was split into chunks. However, even those reasonable-sized chunks were too large and required to generate a separate file containing questions and add them back into the training files. In the future, as LLMs' context windows expand, we may be able to more efficiently explore further.

Another limitation we faced was regarding computing power. Affordable compute power and GPU access are often limited. We were only able to run ten trials per setup with the resources we had, but we aim to extend this analysis in the future.

Acknowledgments

We thank Rachel Hamelburg and Aliyah Vanterpool for providing us code and assistance in designing the prompting process. We thank Dr. Feldman, Dr. Peng and the NLP lab at Montclair State University for helpful discussion and feedback.

References

Igor Bascandziev, Patrick Shafto, and Elizabeth Bonawitz. 2021. The sound of pedagogical questions. In *Proceedings of the annual meeting of the cognitive science society*, volume 43.

BNC Consortium. 2007. The british national corpus, xml edition.

Peter Brodsky and Heidi Waterfall. 2007. Characterizing motherese: On the computational structure of child-directed language. In *Proceedings of the annual meeting of the cognitive science society*, volume 29.

Bastian Bunzeck and Sina Zarrieß. 2023. GPT-wee: How small can a small language model really get? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 35–46, Singapore. Association for Computational Linguistics.

Maureen A Callanan and Lisa M Oakes. 1992. Preschoolers' questions and parents' explanations: Causal thinking in everyday activity. *Cognitive development*, 7(2):213–233.

Thea Cameron-Faulkner, Elena Lieven, and Michael Tomasello. 2003. A construction based analysis of child directed speech. *Cognitive science*, 27(6):843–873.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Emily N Daubert, Yue Yu, Milagros Grados, Patrick Shafto, and Elizabeth Bonawitz. 2020. Pedagogical questions promote causal learning in preschoolers. *Scientific reports*, 10(1):20700.

Baxter S Eaves Jr, Naomi H Feldman, Thomas L Griffiths, and Patrick Shafto. 2016. Infant-directed speech is consistent with teaching. *Psychological review*, 123(6):758.

Lukas Edman and Lisa Bylinina. 2023. Too much information: Keeping training simple for BabyLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 89–97, Singapore. Association for Computational Linguistics.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Akari Haga, Akiyo Fukatsu, Miyu Oba, Arianna Bisazza, and Yohei Oseki. 2024. BabyLM challenge: Exploring the effect of variation sets on language model training efficiency. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 252–261, Miami, FL, USA. Association for Computational Linguistics.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox, editors. 2024. *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Miami, FL, USA.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Anishka Jean, Emily Daubert, Yue Yu, Patrick Shafto, and Elizabeth Bonawitz. 2019. Pedagogical questions empower exploration. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 41.

Jaap Jumelet, Lucas Charpentier, Michael Hu, and Jing Liu. 2025. Babylm_2025. OSF.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Jessica L Montag, Michael N Jones, and Linda B Smith. 2015. The words children hear: Picture books and the statistics for language learning. *Psychological science*, 26(9):1489–1496.

Jessica L Montag, Michael N Jones, and Linda B Smith. 2018. Quantity and diversity: Simulating early word learning environments. *Cognitive science*, 42:375–412.

Aaron Mueller and Tal Linzen. 2023a. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. *arXiv preprint arXiv:2305.19905*.

Aaron Mueller and Tal Linzen. 2023b. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11237–11252, Toronto, Canada. Association for Computational Linguistics.

Mira L Nencheva and Casey Lew-Williams. 2022. Understanding why infant-directed speech supports learning: A dynamic attention perspective. *Developmental Review*, 66:101047.

Elissa L Newport, Henry Gleitman, and Lila R Gleitman. 2020. Mother, i'd rather do it myself. *Sentence first, arguments afterward: Essays in language and learning*, 141.

OpenAI. 2025. Gpt-5 mini. https://openai.com. Lightweight variant of GPT-5 large language model.

Meredith L Rowe. 2008. Child-directed speech: Relation to socioeconomic status, knowledge of child development and child vocabulary skill. *Journal of child language*, 35(1):185–205.

Jessica F Schwab and Casey Lew-Williams. 2016. Repetition across successive sentences facilitates young children's word learning. *Developmental psychology*, 52(6):879.

Catherine E. Snow and Charles A. Ferguson, editors. 1977. *Talking to Children: Language Input and Acquisition*. Cambridge University Press.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Nikitas Theodoropoulos, Giorgos Filandrianos, Vassilis Lyberatos, Maria Lymperaiou, and Giorgos Stamou. 2024. BERTtime stories: Investigating the role of synthetic story data in language pre-training. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 308–323, Miami, FL, USA. Association for Computational Linguistics.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with Bayesian optimization for task-specific word representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 130–139, Berlin, Germany. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Wikimedia. 2023. Simple english wikipedia dump. https://dumps.wikimedia.org/simplewiki/20230301/. Accessed: 2023-07-31.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Guanghao You, Balthasar Bickel, Moritz M Daum, and Sabine Stoll. 2021. Child-directed speech is optimized for syntax-free semantic inference. *Scientific Reports*, 11(1):16527.

Yue Yu, Elizabeth Bonawitz, and Patrick Shafto. 2019. Pedagogical questions in parent–child conversations. *Child development*, 90(1):147–161.

Beyond Repetition: Text Simplification and Curriculum Learning for Data-Constrained Pretraining

Matthew Theodore Roque* **and Dan John Velasco***
Samsung R&D Institute Philippines
{roque.mt,dj.velasco}@samsung.com
*Equal Contribution

Abstract

Most studies on language model pretraining focus on large datasets, leaving open questions about optimization in data-constrained settings. In such settings, the effects of training data order and of including alternative versions of the same text remain underexplored. We address this by studying curriculum learning in pretraining, focusing on text-complexity ordering and data augmentation via simplification. We ask: (1) Does simplifying texts enhance representation quality more than reusing the original data? and (2) Does ordering data by text complexity yield better representations? To answer, we build on a pair of parallel corpora where human-written paragraphs are aligned with LLM-simplified variants, and test four data schedules: repeated exposure, low-to-high complexity, high-to-low, and interleaved. We analyze models' representation quality from a sample efficiency perspective via fine-tuning, as well as its zero-shot performance on linguistic knowledge, entity tracking, world knowledge, and commonsense reasoning. Our findings show that adding simplified data improves fine-tuning and zero-shot performance over a repeated-exposure baseline: smaller models benefit from low-to-high complexity, while larger models perform better with interleaved ordering.

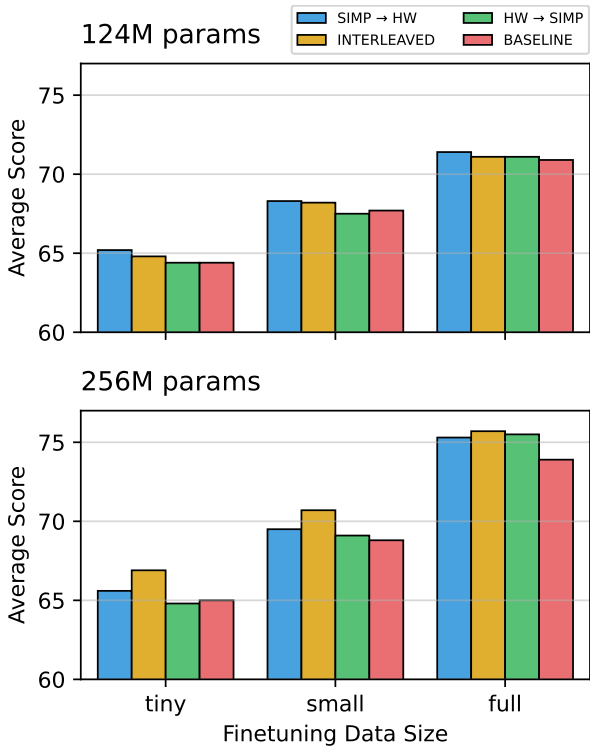


Figure 1: Average score on six language tasks by curriculum and fine-tuning data size. **(Top)** The 124M model benefits from SIMP → HW curriculum, suggesting smaller models gain from warming up on simpler text. **(Bottom)** The 256M model benefits from INTERLEAVED, favoring balanced exposure. All but HW → SIMP outperform BASELINE in sample efficiency.

1 Introduction

Scaling studies show that language model performance improves predictably with more data, parameters, and compute (Kaplan et al., 2020; Hoffmann et al., 2022). However, these studies typically assume that the amount of unique pretraining data is effectively unlimited (Muennighoff et al., 2025). In practice, pretraining often faces data-constrained settings where continued exposure to the same corpus is unavoidable. Under such conditions, two factors remain underexplored in modern decoder-only pretraining: (1) the order in which training data is presented, and (2) the use of simpler versions of the same text.

We examine these factors concretely. For training data order, we focus on **coarse-grained text complexity ordering**—presenting a simple corpus before a complex one—while keeping the core content constant, following the intuition of curriculum learning (Bengio et al., 2009). We define "simple" texts as those that use higher-frequency vocabulary and have shallower syntactic structures.

Proceedings of the First BabyLM Workshop, pages 258–267

Our experiments use a high-quality English dataset where human-written paragraphs are paired with simplified counterparts produced by an LLM. This corpus was introduced and validated in concurrent work (Velasco and Roque, 2025), which demonstrates that simplification reduces surface-level complexity (sentence length, syntactic depth, lexical diversity) while preserving semantic content. Here, we do not revisit corpus construction in detail; instead, we leverage it to test how simplified data and ordering strategies interact under a fixed training budget.

We evaluate four data schedules:

- `BASELINE`: repeated exposure to human-written text.
- `INTERLEAVED`: human-written and simplified paragraphs uniformly mixed.
- `SIMP→HW`: training first on simplified, then human-written text (curriculum).
- `HW→SIMP`: training first on human-written, then simplified text (anti-curriculum).

All other training variables (architecture, tokenizer, context length, optimizer) are held constant across schedules, isolating the effect of text complexity and order.

Research questions. We ask two core questions:

(1) In data-constrained settings, does replacing repeated exposure with simplified text improve representation quality?

(2) Does ordering data by text complexity—simple to complex versus interleaved—yield better downstream and zero-shot performance?

Contributions. Our contributions are threefold:

(1) We provide the first controlled study of how text simplification and curriculum scheduling interact in data-constrained pretraining.

(2) We evaluate these schedules across fine-tuning and zero-shot tasks, covering linguistic knowledge, entity tracking, world knowledge, and commonsense reasoning.

(3) We show that simplified data generally improves performance over repeated exposure, that smaller models benefit from simple-to-complex curricula, and that larger models favor balanced exposure via interleaving.

2 Related Work

Data-constrained Pretraining and Synthetic Data. Most scaling studies assume unlimited data. In data-constrained settings, Muennighoff et al. (2025) shows that training up to four epochs of repeated data is just as good as unique data, but further repetition offers no benefit. Recent works address the "data wall" either by using LLMs to *generate* synthetic data (Gunasekar et al., 2023; Ben Allal et al., 2024) or to *rewrite* existing data, broadly across general domains (Maini et al., 2024; Su et al., 2025; Nguyen et al., 2025; DatologyAI et al., 2025) and in specific domains such as math, code, and clinical text (Fujii et al., 2025; Liu and Nguyen, 2024). Rewriting into simpler forms remains underexplored; we investigate this setting.

Curriculum Learning. Humans learn better when examples follow a meaningful order (e.g., simple to complex). Bengio et al. (2009) formalized this as Curriculum Learning (CL), training neural networks on data of gradually increasing complexity. The opposite, anti-curriculum, has sometimes matched or outperformed CL (Kocmi and Bojar, 2017; Zhang et al., 2018). Difficulty metrics vary by application: in code language models, Naïr et al. (2024) used software engineering metrics, while in machine translation, Zhou et al. (2020) measured sequence uncertainty via language model entropy. Recent studies applied CL to pretraining language models: Tsvetkov et al. (2016) examined linguistically inspired measures such as age of acquisition, while Oba et al. (2023) measured complexity via dependency tree depth. A recent large-scale study by Zhang et al. (2025) finds that using text complexity metrics such as Flesch Reading Ease, Lexical Diversity, and Compression Ratio can accelerate convergence and modestly outperform random shuffling.

However, most CL studies in pretraining conflate language and content complexity, failing to isolate the effect of ordering by language complexity. Most pretraining studies also assume large data volumes, leaving open the question of how to advance pretraining in data-constrained settings beyond repeated exposure. This work uniquely addresses this gap, testing whether LLM-based simplification and coarse-grained text complexity ordering improve representational quality beyond repeated exposure to the original data.

3 Methodology

3.1 Data

We reuse the two parallel corpora from Velasco and Roque (2025), derived from a 2B-token subset of FineWeb-Edu (ODC-By 1.0; Penedo et al., 2024). The human-written corpus (HW) and its simplified variant (SIMP) are aligned at the paragraph level: each paragraph in HW has a corresponding simplification in SIMP produced by an LLM, with paragraphs that fail basic length/formatting checks symmetrically removed from both sides to preserve one-to-one alignment. After filtering, token counts are approximately 2.00B (HW) and 1.71B (SIMP).

In brief, SIMP reduces surface-level complexity (shorter sentences, shallower syntax, more frequent vocabulary) while preserving core content; Velasco and Roque (2025) report readability and lexical/syntactic metrics alongside semantic-similarity checks that validate this property. Our study focuses on how to use these corpora under a fixed budget: repeated exposure vs. augmentation with simplifications and interleaving vs. ordered curricula. All other variables (architecture, tokenizer, context length, optimizer) are held constant across schedules.

3.2 Schedules

We compare four data schedules that differ only in the order of data presented to the model. Unlike most curriculum strategies that manipulate complexity per example (fine-grained), our schedules are coarse-grained, adjusting complexity at the corpus level rather than per example. The four schedules are as follows:

- BASELINE: two epochs of HW (simulating data-constrained scenarios).
- INTERLEAVED: HW and SIMP are uniformly interleaved, preserving each corpus's within-source order (simulating random shuffling in a more balanced way).
- SIMP→HW: concatenation of SIMP and HW (simulating standard curriculum).
- HW→SIMP: concatenation of HW and SIMP (simulating anti-curriculum).

Each training example is a single paragraph, and both HW and SIMP corpora are perfectly parallel, containing the same number of paragraphs. Within each source, paragraph order is fixed across all schedules, ensuring differences arise only from the sequence in which the two sources are presented. This design isolates the effect of data ordering by text complexity and the presence of LLM-rewritten text, while controlling for content coverage and total training steps.

3.3 Model and Training

We use 124M and 256M parameter causal language models based on the design principles of MobileLLM (Liu et al., 2024), adopting a deep-and-thin architecture with SwiGLU activations (Shazeer, 2020), grouped-query attention (Ainslie et al., 2023), and without embedding weight sharing for better comparability with contemporary decoder-only models. Each model has 30 transformer layers with 9 attention heads (3 key–value heads per layer) and embedding dimensions of 576 (124M) and 846 (256M). We refer to the configurations as 124M/256M by convention; the total parameter counts including embeddings are approximately 143M and 283M, respectively.

All corpora are tokenized using the LLaMA-2 BPE tokenizer (Touvron et al., 2023) with a 32,000-token vocabulary. Training examples are individual paragraphs, with no concatenation or sequence packing to control for total training steps. Inputs are right–padded to 512 tokens with the EOS token.

Optimization uses AdamW (Loshchilov and Hutter, 2019) with default hyperparameters, a peak learning rate of $3e{-}4$ linearly decayed over training, 5% warm-up, and no dropout. The effective batch size is 256 (8 examples per GPU × 8 GPUs × 4 gradient accumulation steps). Training is conducted in FP16 mixed precision on 8× NVIDIA P100 GPUs, without gradient checkpointing. All experiments use PyTorch with Hugging Face Transformers and Distributed Data Parallel (DDP), with a fixed random seed of 42 for data shuffling and parameter initialization.

3.4 Evaluation Setup

We evaluate each schedule under two complementary regimes to capture both transferable language understanding after fine-tuning and generalization without further task-specific training.

Fine-tuning on NLU tasks. We fine-tune the pretrained models on a subset of the BabyLM evaluation pipeline's natural language understanding (NLU) tasks, using the preprocessed training and validation splits provided therein (Charpentier et al.,

2025). From the original suite, we exclude:

- **MultiRC**, due to its substantially larger size and because preliminary experiments showed none of our pretraining setups outperformed training from scratch (58–59 points).
- **WSC**, due to its much smaller dataset size and the high variance (4–12 points) observed across seeds under different fine-tuning budgets.

The resulting set of tasks is: BoolQ, MNLI, MRPC, QQP, and RTE. Each task is framed as classification, with paired-input tasks (e.g., premise-hypothesis) concatenating the two sequences with a separator token. Fine-tuning is performed with all model parameters trainable. Batch sizes are set by memory constraints: 2 examples per GPU (effective batch size 16) for BoolQ, and 8 examples per GPU (effective batch size 64) for all other tasks. For each task, we search over learning rates $\{1e{-}4, 5e{-}5, 2e{-}5, 1e{-}5, 5e{-}6\}$ and epochs $\{1, 2, 3, 4, 5\}$. Model selection is based on the highest validation score according to the task's standard metric (accuracy for BoolQ, MNLI, and RTE; F1 for MRPC and QQP). All results are averaged over three runs with different random seeds.

In addition to this **Full** fine-tuning setup, we introduce two smaller-scale, class-balanced variants to test whether having more downstream task examples diminishes the influence of pretraining differences. For each task, we identify the class with the fewest examples in the original training split, then construct:

1. **Small** dataset, in which all classes contain exactly half of the least-represented class size, and
2. **Tiny** dataset, in which all classes contain exactly one quarter of the least-represented class size.

This ensures that all classes are equally represented and that dataset size is systematically reduced across tasks.

These smaller datasets are fixed across runs so that each of the three seeds per setup uses the same subset of examples, with only model initialization differing. The same hyperparameter sweeps are applied to these reduced datasets as in the full-data setup.

Task	Classes	Full	Small	Tiny
MNLI	3	10,000	4,911 (49%)	2,454 (25%)
MRPC	2	3,668	1,194 (33%)	596 (16%)
BoolQ	2	9,427	3,552 (38%)	1,776 (19%)
RTE	2	2,490	1,240 (50%)	620 (25%)
QQP	2	10,000	3,662 (37%)	1,830 (18%)

Table 1: Number of training examples per NLU task in the **Full**, **Small**, and **Tiny** fine-tuning setups. Small and tiny datasets are class-balanced subsets, with percentages shown relative to the full dataset. Percentages vary across tasks because subset sizes are determined relative to the least-represented class in each dataset.

Zero-shot evaluation. Using the LM Evaluation Harness (Gao et al., 2024), we assess models trained under different curricula via zero-shot performance on a suite of multiple-choice benchmarks, grouped into three capability areas:

- **Linguistic knowledge:** BLiMP (Warstadt et al., 2020) and BLiMP Supplement (Charpentier et al., 2025) for syntactic and morphological phenomena.
- **Discourse and world knowledge:** Entity Tracking (Kim and Schuster, 2023) for reference consistency, and MMLU (Hendrycks et al., 2021) and EWoK (Ivanova et al., 2024) for factual knowledge.
- **Commonsense reasoning:** ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), Social IQa (Sap et al., 2019), and OpenBookQA (Mihaylov et al., 2018) for reasoning about everyday scenarios.

All zero-shot tasks are formatted according to their official specifications. For each candidate answer, we compute the sum of log-probabilities of its tokens given the prompt and select the option with the highest score. Accuracy is reported for all tasks. Evaluation is deterministic and performed on a single NVIDIA P100 GPU.

4 Results and Discussion

Across fine-tuning and zero-shot setups, outcomes hinge on two factors: augmenting with simplified data versus repeated exposure, and ordering of data from simplified to complex versus interleaving.

4.1 Simplification vs. Repetition

We ask whether replacing a second pass over the human-written corpus with LLM-based simplifications improves pretraining. The primary

Model	Full	Small	Tiny
124M			
BASELINE	70.9	67.7	64.4
INTERLEAVED	71.1	68.2	64.8
SIMP → HW	**71.4**	**68.3**	**65.2**
HW → SIMP	71.1	67.5	64.4
256M			
BASELINE	73.9	68.8	65.0
INTERLEAVED	**75.7**	**70.7**	**66.9**
SIMP → HW	75.3	69.5	65.6
HW → SIMP	75.5	69.1	64.8

Table 2: Macro-average across five NLU tasks (BoolQ, MNLI, MRPC-F1, QQP-F1, RTE) under three fine-tuning budgets: **Full**, **Small**, and **Tiny**. Best per model size per budget in bold. At 124M, SIMP→HW leads across all budgets (+0.5–0.8 vs. BASELINE); at 256M, INTERLEAVED leads across all budgets (+1.8–1.9), indicating size- and budget-dependent preferences. Per-task breakdowns with mean ± std. dev. are in Appendix A.

comparison is BASELINE (two epochs of HW) vs. INTERLEAVED, which uniformly mixes HW and SIMP.

Fine-tuning evaluation. At 124M, adding SIMP yields small, consistent gains over BASELINE when mixed (INTERLEAVED: +0.2, +0.5, +0.4 on **Full/Small/Tiny**). At 256M, the benefits of including SIMP are larger overall in the mixed setting (INTERLEAVED: +1.8/+1.9/+1.9). These advantages grow as fine-tuning data decreases: INTERLEAVED degrades less than BASELINE on Small and Tiny setups, suggesting simplification is most beneficial when data is scarce, improving sample efficiency.

Zero-shot evaluation. Across linguistic, discourse, and commonsense benchmarks, introducing SIMP tends to be neutral to positive at 124M and more evidently positive at 256M, with the largest single gain appearing on entity tracking when mixing (INTERLEAVED vs. BASELINE: $\approx +4.9$ points at 256M, Table 4). Other tasks show smaller, sometimes mixed, changes, so the aggregate trend favors simplification but with task-family variability (Table 3).

Summary. Compared to repeating HW, in-domain augmentation via LLM simplification yields modest gains at 124M and larger gains at 256M. While we do not claim causality, a plausible explanation is that the additional capacity of the 256M model may better absorb paraphrastic variety, making mixed (INTERLEAVED) exposure yield small but consistent improvements in both fine-tuned and zero-shot settings. We consider this hypothesis specific to our training setup rather than a general rule.

While repetition underperforms simplification, it remains attractive given its zero generation cost and competitive results, consistent with findings by Muennighoff et al. (2025). **Our results complement rather than replace repeated exposure: since performance saturates beyond four epochs, text simplification offers a way to extend these gains further in data-constrained settings.**

4.2 Curriculum vs. Interleaving

We next ask whether ordering simplified and human-written text into two distinct phases (curriculum or anti-curriculum) provides advantages over mixing them uniformly. Here, INTERLEAVED serves as the natural baseline, representing random shuffling of the two corpora.

Fine-tuning evaluation. At 124M, the curriculum schedule SIMP→HW edges out INTERLEAVED across all budgets (+0.3, +0.1, +0.4), while HW→SIMP is equal or slightly worse (Table 2). The margins are small, but the consistent advantage for SIMP→HW suggests that a warm-up on simplified text may provide downstream benefits when model capacity is limited. At 256M, the picture changes: INTERLEAVED remains strongest across all budgets, while the ordered schedules are competitive but do not surpass it (SIMP→HW: -0.4, -1.2, -1.3 vs. INTERLEAVED; HW→SIMP: -0.2, -1.6, -2.1). In short, simple-to-complex ordering helps at 124M, especially in low-resource settings where it shows improved sample efficiency, but offers no advantage as model capacity increases.

Zero-shot evaluation. For zero-shot tasks, ordering effects are somewhat stronger than in the fine-tuning results. At 124M, SIMP→HW leads on BLiMP and BLiMP-Supplement, while HW→SIMP excels on entity tracking (Table 4). At 256M, however, INTERLEAVED typically matches or exceeds the ordered setups—for example, it leads on entity tracking (+4.3 vs. SIMP→HW) and holds a small edge on HellaSwag (Table 3). These patterns suggest that ordered exposure can steer smaller models toward specific strengths, but that random mixing is safer once capacity is sufficient.

Summary. Compared to interleaving, SIMP → HW yields modest gains at 124M—especially for NLU fine-tuning and linguistic probes—but these benefits diminish or reverse at 256M. This pattern

Model	arc_chl	arc_e	hellaswag	openbookqa	piqa	social_iqa	Avg.
124M							
BASELINE	22.0	**35.9**	**28.4**	14.0	57.1	36.0	34.28
INTERLEAVED	**23.4**	34.6	**28.4**	15.4	**58.4**	35.7	**34.50**
SIMP → HW	23.1	33.7	28.0	**17.0**	57.3	36.1	34.42
HW → SIMP	20.4	33.5	28.1	14.8	57.6	**36.9**	34.18
256M							
BASELINE	23.1	**37.8**	27.7	16.0	57.3	35.2	34.80
INTERLEAVED	22.7	36.9	28.6	16.2	56.8	**36.6**	35.02
SIMP → HW	21.8	35.5	**29.1**	**18.2**	**58.4**	36.5	**35.54**
HW → SIMP	**24.7**	34.7	28.2	16.6	56.4	35.5	34.28

Table 3: Zero-shot accuracy on commonsense reasoning benchmarks (ARC-Challenge, ARC-Easy, HellaSwag, OpenBookQA, PIQA, Social IQa). "Avg." is the mean across tasks. At 124M, INTERLEAVED yields the highest average by a small margin; at 256M, SIMP → HW attains the best average, driven by gains on HellaSwag, OpenBookQA, and PIQA, while INTERLEAVED leads on Social IQa and HW → SIMP peaks on ARC-Challenge.

Model	blimp	supp	ewok	entity	mmlu
124M					
BASELINE	71.8	61.9	53.9	22.4	24.7
INTERLEAVED	72.3	63.6	**55.5**	28.1	**24.9**
SIMP → HW	**72.4**	**63.8**	54.8	31.7	23.6
HW → SIMP	70.7	61.3	55.1	**36.9**	23.3
256M					
BASELINE	**73.8**	**65.6**	55.0	30.1	23.5
INTERLEAVED	73.6	64.1	**56.2**	**35.0**	24.5
SIMP → HW	73.7	64.3	55.9	30.8	25.8
HW → SIMP	73.1	64.3	56.0	34.1	**26.2**

Table 4: Zero-shot evaluations on linguistic competence (BLiMP, BLiMP-Supplement), world knowledge (EWoK), discourse (Entity Tracking), and general-domain reasoning (MMLU). For 124M models, SIMP → HW leads on BLiMP, while HW → SIMP achieves the highest score on Entity Tracking. At 256M, performance differences are narrower, with small trade-offs across tasks.

suggests that smaller models benefit from a simple-to-complex progression in surface-level complexity, whereas larger models can internalize both variants without explicit ordering. **We view these as hypotheses under our training regime: ordering appears to matter more in low-capacity settings and is less critical when models scale up.**

5 Conclusion

This work examines whether augmenting pretraining data with LLM-based simplifications and ordering by text complexity improves representation quality in data-constrained settings. Through controlled data and training conditions, we isolated the effects of text complexity in curriculum design across two model sizes.

Our findings suggest three takeaways:

(1) Adding simplified data outperforms repeating human-written text, yielding modest gains at 124M and clearer benefits at 256M.

(2) Curriculum effects depend on model scale: smaller models benefit slightly more from a simple-to-complex curriculum, while larger models favor balanced exposure via interleaving.

(3) Differences are most evident in zero-shot and low-resource fine-tuning scenarios, where schedule choice impacts representational quality, improving zero-shot performance and sample efficiency in small fine-tuning budgets.

Overall, our results show that in data-constrained settings, text simplification and curriculum learning can complement repeated exposure. Although the gains are modest, they suggest practical ways to extend the utility of pretraining without fresh data collection. Future work includes testing longer training horizons, exploring rewriting methods beyond simplification, and evaluating schedule effects at larger scales and in non-English domains.

Limitations

Our study is intentionally narrow in scope. We test only two model sizes (124M and 256M) with a single decoder-only architecture, leaving open how schedule effects scale to larger models or alternative designs. Training is constrained to a fixed budget of roughly 4B tokens (two full passes over a 2B-token human-written corpus), so results may differ under longer horizons or larger-scale training. We also study only one form of rewriting

(LLM-based simplification) while other transformation types (e.g., paraphrasing, elaboration, style transfer) may behave differently. Finally, the evaluation focuses on educational data in English and a limited set of NLU and zero-shot benchmarks, the results may not transfer directly to other domains, languages, or broader task families.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 41–48. ACM.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

DatologyAI, :, Pratyush Maini, Vineeth Dorna, Parth Doshi, Aldo Carranza, Fan Pan, Jack Urbanek, Paul Burstein, Alex Fang, Alvin Deng, Amro Abbas, Brett Larsen, Cody Blakeney, Charvi Bannur, Christina Baek, Darren Teh, David Schwab, Haakon Mongstad, and 12 others. 2025. Beyondweb: Lessons from scaling synthetic data for trillion-scale pretraining. *Preprint*, arXiv:2508.10975.

Kazuki Fujii, Yukito Tajima, Sakae Mizuki, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Masanari Ohi, Masaki Kawamura, Taishi Nakamura, Takumi Okamoto, Shigeki Ishida, Kakeru Hattori, Youmi Ma, Hiroya Takamura, Rio Yokota, and Naoaki Okazaki. 2025. Rewriting pre-training data boosts llm performance in math and code. *Preprint*, arXiv:2505.02881.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *Preprint*, arXiv:2306.11644.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Kathryn Millican, Bogdan Damoc, Arthur Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, and 2 others. 2022. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyurek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.

Jinghui Liu and Anthony Nguyen. 2024. Rephrasing electronic health records for pretraining clinical language models. In *Proceedings of the 22nd Annual Workshop of the Australasian Language Technology Association*, pages 164–172, Canberra, Australia. Association for Computational Linguistics.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. *Preprint*, arXiv:2402.14905.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14044–14072, Bangkok, Thailand. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2025. Scaling data-constrained language models. *Preprint*, arXiv:2305.16264.

Marwa Naïr, Kamel Yamani, Lynda Lhadj, and Riyadh Baghdadi. 2024. Curriculum learning for small code language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 390–401, Bangkok, Thailand. Association for Computational Linguistics.

Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. 2025. Recycling the web: A method to enhance pre-training data quality and quantity for language models. *Preprint*, arXiv:2506.04689.

Miyu Oba, Akari Haga, Akiyo Fukatsu, and Yohei Oseki. 2023. BabyLM challenge: Curriculum learning based on sentence complexity approximating language acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 290–297, Singapore. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2025. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2459–2475, Vienna, Austria. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with Bayesian optimization for task-specific word representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 130–139, Berlin, Germany. Association for Computational Linguistics.

Dan John Velasco and Matthew Theodore Roque. 2025. Rethinking the role of text complexity in language model pretraining. *arXiv preprint arXiv:2509.16551*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine

Carpuat. 2018. An empirical exploration of curriculum learning for neural machine translation. *Preprint*, arXiv:1811.00739.

Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. 2025. Beyond random sampling: Efficient language model pretraining via curriculum learning. *Preprint*, arXiv:2506.11300.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

A Per-task Results of Fine-tuning Evaluation

Results are presented by fine-tuning budget: Full (Table 5), Small (Table 6), and Tiny (Table 7).

Model	BoolQ	MNLI	MRPC	QQP	RTE	Avg.
124M						
BASELINE	70.6 ± 0.3	66.2 ± 0.7	82.2 ± 1.7	80.9 ± 0.3	66.4 ± 3.1	70.9
INTERLEAVED	69.5 ± 0.6	67.5 ± 0.2	80.6 ± 1.0	81.1 ± 0.3	69.2 ± 2.4	71.1
SIMP → HW	70.7 ± 0.6	67.1 ± 0.7	82.2 ± 2.2	81.3 ± 0.1	68.1 ± 2.8	71.4
HW → SIMP	71.2 ± 0.7	66.7 ± 0.7	83.3 ± 0.7	80.6 ± 0.1	66.2 ± 1.7	71.1
256M						
BASELINE	71.2 ± 0.1	69.2 ± 0.5	81.9 ± 1.1	81.5 ± 0.1	65.7 ± 3.1	73.9
INTERLEAVED	71.0 ± 0.3	69.8 ± 0.1	84.1 ± 1.7	82.2 ± 0.1	71.5 ± 1.8	75.7
SIMP → HW	71.0 ± 0.8	70.3 ± 0.3	82.9 ± 1.0	82.0 ± 0.1	70.1 ± 2.5	75.3
HW → SIMP	71.8 ± 0.2	69.5 ± 0.3	82.2 ± 1.0	82.0 ± 0.2	71.8 ± 2.6	75.5

Table 5: Per-task fine-tuning results under **Full** fine-tuning budget. "Avg." is the mean across tasks. Results are grouped by model size (124M and 256M) and curriculum strategy.

Model	BoolQ	MNLI	MRPC	QQP	RTE	Avg.
124M						
BASELINE	62.9 ± 1.0	63.1 ± 0.4	70.8 ± 1.0	78.1 ± 0.9	63.4 ± 4.5	67.7
INTERLEAVED	62.4 ± 0.3	64.0 ± 0.3	72.4 ± 1.0	78.7 ± 0.0	63.7 ± 1.6	68.2
SIMP → HW	64.1 ± 0.6	63.1 ± 0.3	71.5 ± 0.7	79.0 ± 0.2	63.9 ± 0.7	68.3
HW → SIMP	62.7 ± 0.7	63.1 ± 0.2	73.2 ± 0.3	78.3 ± 0.4	60.2 ± 3.3	67.5
256M						
BASELINE	63.9 ± 0.7	65.1 ± 0.2	72.8 ± 2.3	79.3 ± 0.1	63.2 ± 2.4	68.8
INTERLEAVED	63.7 ± 0.6	65.8 ± 0.5	73.1 ± 1.7	80.1 ± 0.2	71.1 ± 2.8	70.7
SIMP → HW	64.2 ± 0.3	66.1 ± 0.7	71.5 ± 1.7	79.7 ± 0.1	66.0 ± 0.7	69.5
HW → SIMP	63.7 ± 0.2	65.4 ± 0.7	71.6 ± 3.8	79.4 ± 0.2	65.3 ± 4.9	69.1

Table 6: Per-task fine-tuning results under **Small** fine-tuning budget. "Avg." is the mean across tasks. Results are grouped by model size (124M and 256M) and curriculum strategy.

Model	BoolQ	MNLI	MRPC	QQP	RTE	Avg.
124M						
BASELINE	59.5 ± 0.7	59.4 ± 1.0	67.9 ± 0.3	75.3 ± 0.4	59.7 ± 6.1	64.4
INTERLEAVED	58.4 ± 0.4	60.7 ± 0.3	70.4 ± 1.5	76.0 ± 0.2	58.3 ± 1.2	64.8
SIMP → HW	58.8 ± 1.0	59.8 ± 0.4	68.9 ± 2.6	76.5 ± 0.1	62.0 ± 0.8	65.2
HW → SIMP	59.5 ± 1.5	60.3 ± 0.1	71.6 ± 0.8	75.6 ± 0.5	55.1 ± 3.3	64.4
256M						
BASELINE	60.8 ± 1.3	60.7 ± 0.4	68.1 ± 1.2	76.5 ± 0.6	59.0 ± 1.8	65.0
INTERLEAVED	60.1 ± 0.7	61.6 ± 0.5	71.0 ± 1.9	77.5 ± 0.2	64.6 ± 0.7	66.9
SIMP → HW	60.5 ± 1.0	60.9 ± 0.9	68.6 ± 1.8	76.9 ± 0.3	61.1 ± 2.5	65.6
HW → SIMP	59.9 ± 0.6	61.4 ± 0.4	65.5 ± 4.1	76.2 ± 0.3	60.9 ± 4.2	64.8

Table 7: Per-task fine-tuning results under **Tiny** fine-tuning budget. "Avg." is the mean across tasks. Results are grouped by model size (124M and 256M) and curriculum strategy.

CurLL: A Developmental Framework to Evaluate Continual Learning in Language Models

Pavan Kalyan
Microsoft Research
tankalapavankalyan@gmail.com

Shubhra Mishra
KTH Royal Institute of Technology
Stockholm, Sweden
shubhram@kth.se

Satya Lokam
Microsoft Research
satya.lokam@microsoft.com

Navin Goyal
Microsoft Research
navingo@microsoft.com

Abstract

We introduce a comprehensive continual learning dataset and benchmark (CURLL) grounded in human developmental trajectories from ages 5–10, enabling systematic and fine-grained assessment of models' ability to progressively acquire new skills. CURLL spans five developmental stages (0–4) covering ages 5–10, with a skill graph of 32 high-level skills, 128 sub-skills, 350+ goals, and 1,300+ indicators explicitly modeling prerequisite relationships. We generate a 23.4B-token synthetic dataset with controlled skill progression, vocabulary complexity, and format diversity, comprising paragraphs, comprehension-based QA (CQA), skill-testing QA (CSQA), and instruction–response (IR) pairs. Stage-wise token counts range from 2.12B to 6.78B tokens, supporting precise analysis of forgetting, forward transfer, and backward transfer. Using a 135M-parameter transformer trained under independent, joint, and sequential (continual) setups, we show trade-offs in skill retention and transfer efficiency. By mirroring human learning patterns and providing fine-grained control over skill dependencies, this work advances continual learning evaluations for language models.

1 Introduction

The ability to continuously learn and adapt to new information throughout life is one of the hallmarks of human intelligence. Unlike current artificial intelligence systems (e.g., LLMs, agents), humans integrate new knowledge with existing understanding, build increasingly complex skills on earlier foundations, and retain previous capabilities even as they master new ones, and achieve all this with very high sample efficiency. This capacity for lifelong learning represents not just a practical advantage but a fundamental aspect of intelligence itself (Kudithipudi et al., 2022; Yan et al., 2024; Schmidgall et al., 2023).

The continual learning (CL) problem thus is one of the grand challenges for achieving human-like artificial intelligence. It addresses the core problem of how computational systems can progressively acquire, integrate, and refine knowledge over extended periods without compromising earlier capabilities. For language models (LMs), this challenge is particularly interesting: despite their impressive performance across various tasks, these models face a fundamental limitation in that their skill-set and knowledge of the world becomes static after training, frozen at the point of deployment (Shi et al., 2024; Wu et al., 2024; Bell et al., 2025). In real world, this information continually expands and updates, and this limitation poses a significant challenge to the long-term utility and relevance of LMs.

Despite the importance of the continual learning problem for LMs, current evaluation methodologies suffer from significant limitations:

1. Poor skill control: Existing benchmarks often lack precise control over the specific skills being tested, making it difficult to isolate the effects of learning new capabilities (Liu et al., 2025a; Rivera et al., 2022).

2. Unclear knowledge dependencies: The relationships between different skills are rarely explicitly modeled, thus missing out on important transfer effects (Zheng et al., 2025; Nekoei et al., 2021).

3. Inadequate forgetting metrics: Many evaluations fail to properly measure catastrophic forgetting across sequential learning tasks (Chen et al., 2023a; Huang et al., 2023).

These limitations make it difficult to understand to measure the efficacy of continual learning algorithms for LMs. This in turn impedes the development of more effective algorithms.

To address these gaps, we introduce a dataset (CURLL) to train and evaluate continual learning

Proceedings of the First BabyLM Workshop, pages 268–290

algorithms for language models. Coming up with a set of skills with a rich structure and dependencies is a challenge in the construction of such a dataset. We find such a source of skills in human education. (CURLL) is grounded in the curriculum for human education from ages 5–10, divided into five developmental stages (0–4). Each of these stages represent one human-year. Our framework incorporates 1,300+ fine-grained skills. The dependencies among these skills are codified in a skill graph having skills as nodes with the edges capturing a prerequisite relationship. The edges are weighted on a scale of (1–5) to capture dependency strength. Starting from this set of skills, we generate a synthetic dataset of 23.4B tokens, with controlled vocabulary complexity (stage-specific word sampling from Age-of-Acquisition data as seed) and multiple formats (paragraphs, comprehension QA, skill-testing QA, instruction–response). Each stage's dataset ranges from 2.12B to 6.78B tokens, enabling fine-grained evaluation at indicator, skill, and stage levels. Our code, dataset (stages 0–4), and skill graph will be publicly released. Our contributions include:

- The idea of grounding skills in human education curriculum in the context of continual learning
- A synthetic data generation pipeline spanning 5 developmental stages with stage-specific vocabulary, multi-format outputs, and explicit skill dependencies
- This pipeline gives us a benchmark with fine-grained control over measuring skill transfer, forgetting and sample efficiency
- A skill graph-based dependency model that explicitly captures prerequisite relationships between learning objectives, enabling nuanced analysis of skill transfer and forgetting

2 Related Work

One particular limitation of LMs is that their knowledge is confined to fixed parameters established during training (Du et al., 2023). While LLMs encode world knowledge in their parameters through pretraining, this knowledge can quickly become outdated as the world changes (Jang et al., 2021a). Continual learning techniques address this by enabling models to learn continually and adapt, integrating new knowledge and skills while retaining previously learned information (Zheng et al., 2024). This capability represents a fundamental property of human intelligence: the capacity to dynamically adapt cognition by ingesting new knowledge from the environment over time (Du et al., 2023; Jin et al., 2021). The core challenge in continual learning is the stability-plasticity dilemma, which requires models to balance the previous skills (stability) with the ability to learn new tasks (plasticity) (Jiang et al., 2024; Wang et al., 2025; Liu et al., 2025b). Catastrophic forgetting emerges as the primary manifestation of this challenge, where LMs tend to forget previously acquired knowledge when learning new instances (Huang et al., 2024; Zeng et al., 2023; Liao et al., 2025).

Many datasets and benchmarks exist for continual learning of language models such as TRACE (Wang et al., 2023), MMLM-CL (Zhao et al., 2025), OCKL (Wu et al., 2023), CKL (Jang et al., 2021b), TemporalWiki (Jang et al., 2022) and TiC-LM (Li et al., 2025) etc. TRACE (Wang et al., 2023) highlights the problem in existing benchmarks that are often too simple or are already included in the LLM instruction-tuning sets. It also introduces new metrics to evaluate shift in LLM abilities. MMLM-CL (Zhao et al., 2025) discusses the shortcomings of current Cl benchmarks as lack of real world applicability and IID evaluation. OCKL (Wu et al., 2023) proposes new metrics for measuring knowledge acquisition rate and knowledge gap but concentrates primarily on knowledge-intensive tasks as compared to procedural tasks. TemporalWiki (Jang et al., 2022) also concentrates on updating factual information in language models based on temporal data constructed from Wikipedia snapshots. Several domain-specific benchmarks exist as well for language models. Continual relation extraction has been evaluated on datasets including Continual-FewRel, Continual-SimpleQuestions, and Continual-TACRED, where relations are partitioned into sequential tasks (Wu et al., 2021). SuperNI contains a variety of traditional NLP tasks and serves as a practical benchmark for continual learning of large language models (He et al., 2024). Yang et al. (2024) introduced the Life Long Learning of LLM (5L-bench) benchmark, which encompasses a curated dataset of question-answer pairs and evaluation metrics for both open-book and closed-book settings.

Despite these developments, existing continual learning benchmarks are often considered unsuitable for evaluating state-of-the-art LMs (Wang

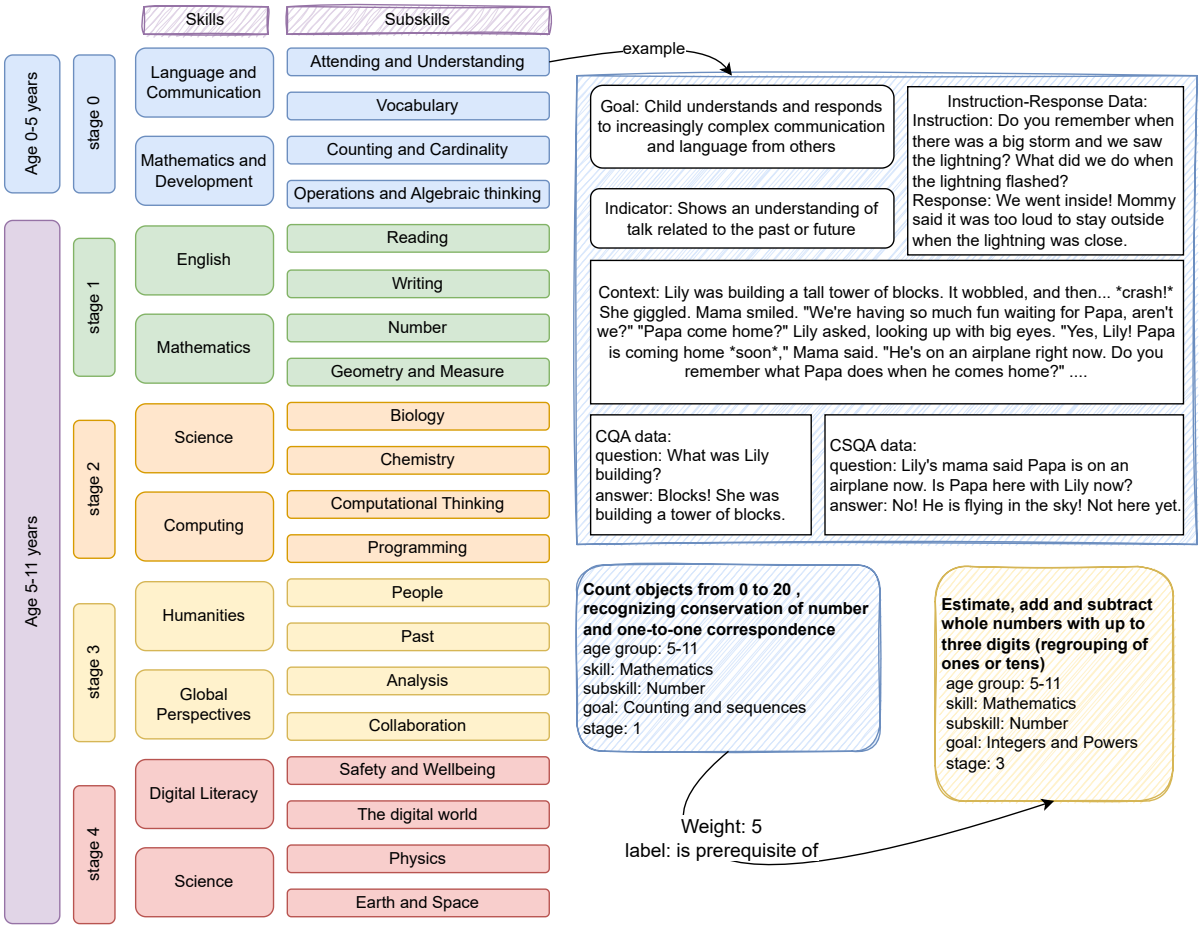


Figure 1: Developmental framework for children aged 0-11 years, categorized into stages (0-4). Only examples of skills and subskills are mentioned here. An example of how the data looks like is given in the top right. Two nodes and an edge from the skill graph is given in the bottom right.

et al., 2023; Razdaibiedina et al., 2023; Scialom et al., 2022; Zhang et al., 2015). These benchmarks often emphasize artificial task boundaries (He et al., 2024), lack temporal and distributional complexity. Moreover, these datasets do not offer precise control over skills or information to validate the effectiveness of existing solutions for continual learning. Skill-it (Chen et al., 2023b) introduces a data sampling algorithm for continual pretraining and finetuning. They do this by arranging the skills in a increasing order of complexity. Other existing works (Khetarpal et al., 2020; Greco et al., 2019; Xu et al., 2024) discuss the importance of skill distinction and its effect on evaluating continual learning.

3 Dataset Setup

One of the main design decisions in the construction of our dataset is to precisely specify the skills that the model learns at each stage of continual learning. To this end, our framework for evaluating continual learning is grounded in human learning curriculum, with the dataset designed to mimic the developmental stages from age 5-14. This section details our methodology for constructing the dataset, developing the skill graph that models dependencies between skills, and creating test-train splits for evaluation.

3.1 Grounding in Human Curricula

We use two established educational frameworks to develop our skill taxonomy: the Early learning Outcomes framework (ELOF) for children below age 5 and the Cambridge curriculum for children aged 5-14. These frameworks help us define fine grained notion of skills. this is specified by a skill-tuple which consists of four components:

- Skills[1]: High-level domains or subjects (e.g.

[1]The word skill here has a specific meaning, which is related but not the same as the general notion of skill we have been discussing

Mathematics, Social and Emotional Development)

- Sub-skills: Specific components within a skill (e.g., Counting and Cardinality, Relationship with adults)
- Goals: Broad statement of learning expectations within a sub-skill
- Indicators: Specific, observable behaviors that demonstrate mastery of a goal

Examples of each of these can be seen in Figure 1.

The ELOF framework, introduced by the U.S. Office of Head Start in 2015, provides a comprehensive roadmap for child development from birth to age five across five broad areas: Approaches to Learning, Social and Emotional Development, Language and Literacy, Cognition, and Perceptual, Motor, and Physical Development. For ages 5-11, we use the Cambridge Primary Curriculum, which covers subjects including English, Mathematics, Science, Computing, and Global Perspectives. The curriculum structure flows from subjects (renamed as skills in our framework) to domains/strands (renamed as subskills), then to substrands (goals), each with specific learning objectives (indicators). For children aged 11-14, we use the Cambridge Lower Secondary Curriculum, which maintains the same subjects as the previous age group but has increased complexity with different subskills, goals, and indicators. We also adopt the notion of stages from the Cambridge curriculum in our framework, where each stage corresponds to one year starting from age 5. Therefore, we have 10 stages in our framework, where stage 0 denotes ages up to 5, stage 1 denotes age 5-6 and so on. In this work, we only use stages 0, 1, 2, 3, 4, i.e. until age 9-10 years for data generation and experiments. The number of skill-tuples in our framework is same as the number of indicators present up to stage 4, statistics of which are mentioned in Table 1.

3.2 Skill Graph

A critical component of our framework is the skill graph, which captures the prerequisite relationships between indicators. This is a directed graph that has indicators as nodes, with edges representing prerequisite relationships weighted from 1-5 to indicate dependency strength. These relationships model how skills are built on each other in developmental stages. We use an LLM[2] to predict these dependency relationships between indicators. While the skill graph isn't directly used for skill data generation, it provides insights for analyzing continual learning patterns and interpreting evaluation results. To verify the validity of skill graph, we analyze the distribution of incoming and outgoing edges across different stages, confirming that lower stages generally have fewer incoming dependencies while higher stages have more prerequisite relationships. Figure 2 shows this analysis.

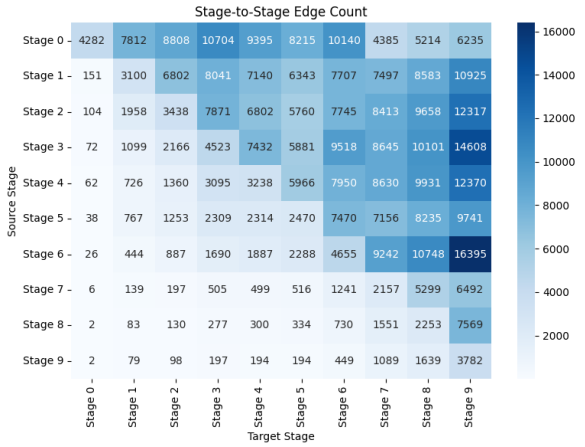


Figure 2: Heatmap showing the number of prerequisite edges between stages in the skill graph. Rows correspond to source stages, columns to target stages, and color intensity indicates the number of connections.

3.3 Synthetic Data Generation

Our synthetic data consists of instances, each mimicking a situation a child might encounter. Instances are of three types: (1) IR: an instruction-response pair, where the instruction is about some general world knowledge, (2) CQA: context-based question-answers for testing comprehension, (3) CSQA: context-based question-answers for testing skills. A context is a short piece of text which forms the basis of the corresponding question-answer pairs in the instance. Contexts can be of multiple types as specified by a template: for example, a simple narrative, or a dialogue. Similarly, IR-pairs can also have different types specified by templates, e.g., mimic action or follow simple direction. See Figure 4 for more examples.

Instances are generated by prompting an LLM with a *seed*. A seed consists of a skill-tuple, vocabulary seed, instance type, template. This choice is

[2]Gemma3-27B-IT is used for all LLM inferences throughout this work

Stage	Skills & Goals				Instances			Total # Tokens in Bn
	# Skills	# Sub-skills	# Goals	# Indicators	# CQA	# CSQA	# IR Pairs	
0	7	24	59	182	1.0M	3.01M	3.30M	2.12
1	7	29	86	292	20.2M	4.04M	4.10M	3.47
2	6	26	67	249	23.5M	4.70M	4.78M	4.56
3	6	26	68	271	31.2M	6.24M	6.29M	6.47
4	6	23	70	349	27.4M	5.49M	5.52M	6.78

Table 1: Dataset statistics across developmental stages (0–4), including generated instances, and total tokens

crucial for ensuring diversity and coverage of our data. An example of our seed and the generated instance is given in Figure 4. This tuple is also our way to ground the generations in the skill graph. In more detail, a seed is generated as follows:

1. Age-appropriate skill grounding: Each generated instance is tied to a specific skill-tuple from our curriculum framework. Since this tuple contains the stage and age group, the generated data is expected to be grounded in the same.

2. Vocabulary seed: To generate the data at scale, we use additional seeds for diversity. One of them is the words from vocabulary of a child belonging to a stage in the curriculum. We do this by using the Age-of-Acquisition data (Kuperman et al., 2012), where words along with the age-rating based on human studies are presented. 1000 words are sampled for each stage. A vocabulary seed consists of one randomly chosen word from this list.

3. One of the instance types (IR, CQA, CSQA).

4. Templates: For each skill-tuple we generate at least 15 sample templates for contexts and for IR-pairs. Therefore each skill tuple has at least 15 types of context templates like stories, dialogue etc. and 15 types of instruction-response templates like why questions, describing the event etc. Examples of these templates are mentioned in Figure 4. We use an LLM to generate these templates by giving the skill-tuple as the input. The prompts for generating context and IR templates are given in the Appendix (A.5.3).

To generate one instance of the data, we first construct a seed: each skill-tuple is combined with a vocabulary seed for that stage, an instance type and a template for that instance type. If the instance type is CQA or CSQA, then we first generate the context and then using the context we generate the corresponding question-answers. If the instance type is IR then we directly generate the instruction-response pairs. The prompts for all the generations are presented in the appendix A.5. In our dataset, each instance includes the seed used to generate it as part of its metadata.

3.4 Data Statistics and Verification

We generated data for stages 0 through 4 inclusive, containing a total of 23.4B tokens (Table 1). We use two methods to measure this diversity of generated data: 1) Diversity as reciprocal of compression ratio using gzip (Gailly and Adler, 1992). 2) The intra- and inter-text deduplication rate as calculated by semantic deduplication. Details of how these measures are implemented are given in the appendix A.1. Cross-stage analysis shows higher diversity and lower deduplication rate (<5%) between stages compared to intra-stage results, confirming that content evolves meaningfully across developmental progression while maintaining stage-specific uniqueness. The results for these methods are presented in Table 2.

Stage	Context		IR	
	Div ↑	Dedup ↓	Div ↑	Dedup ↓
0	34.29%	11.83%	30.77%	3.50%
1	35.60%	5.36%	31.73%	3.85%
2	34.17%	15.47%	32.64%	2.54%
3	34.68%	14.86%	32.97%	2.09%
4	35.45%	13.41%	33.14%	1.93%

Table 2: Diversity and Deduplication metrics for context and instruction–response data across stages

Another important feature of the dataset is the progression in the difficulty of the skills as the stage number increases. We sample 500K instances from each stage for each data type and run statistical readability tests[3]. Means across multiple readabil-

[3]These tests use pre-defined word corpuses to predict the grade a text belongs to. We use the following repo to measure the readability: https://github.com/cdimascio/py-readability-metrics

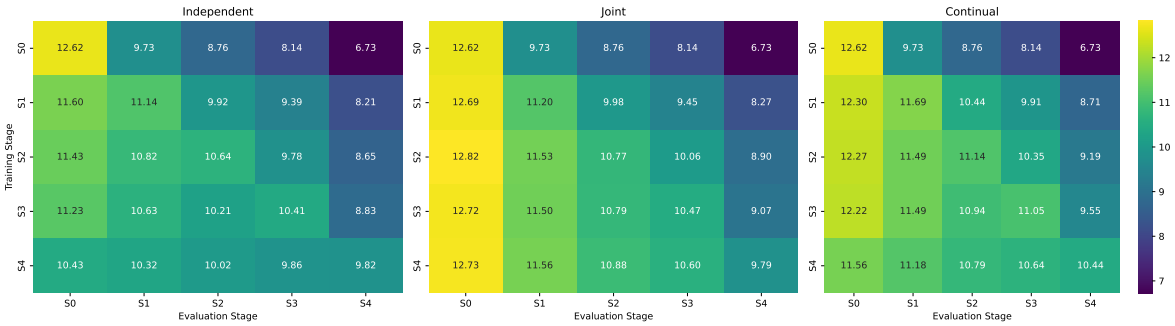


Figure 3: Stage-wise evaluation results for different training setups. Independent training corresponds to models trained on a single stage, joint training to models trained on mixtures of data up to a stage, and continual training to sequential upto a stage. Heatmaps report summed correctness scores across all test formats (IR, CQA, CSQA)

ity metrics are reported in Table 3. The readability tests show that as stages progress, the texts also become increasingly challenging. At least 50 random

Stage	Context	CQA	CSQA	IR
0	4.61 1.87	2.38 2.88	3.07 2.26	4.48 1.52
1	5.24 1.72	4.39 1.81	4.44 1.62	4.86 1.41
2	5.18 1.93	4.39 1.80	4.69 1.54	4.69 1.59
3	5.51 1.85	4.65 1.70	4.98 1.46	5.03 1.50
4	6.42 1.79	5.63 1.44	5.96 1.30	5.91 1.34

Table 3: Average readability scores of generated data across stages, reported for context, comprehension QA (CQA), skill-testing QA (CSQA), and instruction–response (IR) data. Scores generally increase with stage, reflecting controlled growth in textual complexity aligned with developmental progression

instances from each dataset per stage were manually analysed. Based on manual inspection, CQA data for all stages was found to be accurate. IR and CSQA data had certain patterns like excessive use of discourse markers for early stages and verbose response to instructions.

3.5 Train-test split

We aim to keep the size of the test set per stage to be between 5k-7k samples. And to ensure uniform coverage of indicators in the test data, we choose 25 instances per indicator. Since the data is synthetically generated at scale, though we cannot validate the entire data, we reserve the highest quality samples for the test set. Three types of test sets are created based on the three instance types. 100 random samples are selected per indicator for each of the data format. All 100 questions are graded by the LLM on a scale of 1-5 for correctness of the response. The detailed rubrics and prompts are

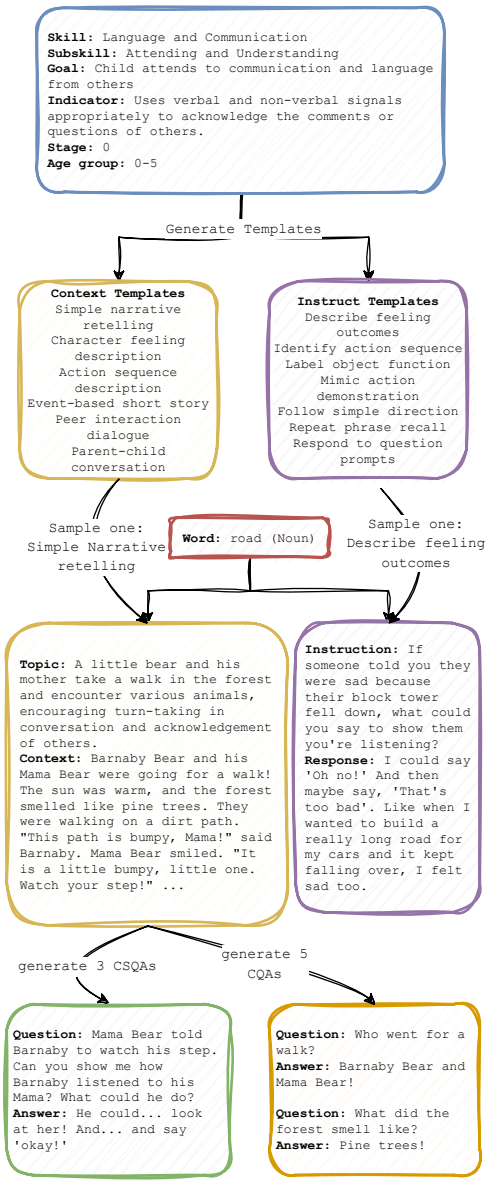


Figure 4: Synthetic data generation pipeline

presented in the appendix A.5. 25 highest scoring (mostly 5) instances out of all instructions were selected in the test set. 25 randomly selected instances from the remaining samples were put in the validation split. And all the rest of the instances remain in the training set. Note that this construction ensures that the test, validation and training sets are disjoint because they use distinct seeds.

4 Experiments and Results

We do preliminary experiments to validate the quality of our dataset, in addition to the checks performed in Section 3.4. Unlike traditional language model training that includes two stages: pretraining and then finetuning, we do a single phase training. All instance types i.e. CQA, CSQA, and IR are included in the same phase. Since all of them are question-answers, with and without context paragraphs, we use a standard chat template to train the language models from scratch. Smollm2-135M parameter model is used as the base architecture. All training runs are performed on one full epoch of the data. Learning rate of 5e-3 and effective batch size of 1536 instances remain unchanged across experiments. We use a context length of 1024. Other training and inference related hyper-parameters are mentioned in the appendix A.2.

Our preliminary experiments includes three types of training:

1. Independent: The model is trained from scratch on data of each stage independently. The models trained in this fashion are denoted by M_i if trained on data of stage i.

2. Joint: The model is jointly trained on a mixture of stages. The data from different stages is combined and shuffled randomly. These models are denoted as M_{ij} if the model is jointly trained on data of stage i and stage j.

3. Continual: The model is first trained on stage i, then stage j, then stage k and so on. This model is denoted by M_{i-j} if it is trained until stage j, by M_{i-j-k} if it is trained until stage k and so on.

4.1 Results

To evaluate the trained models, the instances from test set are passed through the chat template and the model is asked to complete the generation post instruction. These inferences along with the prompt is passed to an LLM to rate on a scale of 1-5. This is followed for all three types of test sets. Each model is evaluated on test sets of all stages. The prompt and rubrics of evaluation are mentioned in the appendix A.5. The main objective of the rating is to evaluate the correctness of the model inference with some weightage to the stage on which the model is being evaluated. The summation of scores across test set types (IR, CQA, CSQA) is presented in Figure 3. The individual scores are available in the appendix.

The Y-axis of the figure shows the stage on which the model is being evaluated. The Y-axis denotes the data used to train the model. For the case of independent training, S_i denotes data from stage i is used for training. For Joint training S_i denotes mixture of data from all stages until i including stage i. For Continual training, this means model trained sequentially until stage i.

The figure shows that as compared to independent models (M_i), joint models(M_{ij}) show better generalization to later stages but also, stronger performance on trained stages. However continual models (M_{i-j}) shows the best performance on later stages but the performance degrades on already trained stages. This is better shown in Figure 5 across different test set types. Even though M_{ij} and M_{i-j} are trained on exactly on the same amount of data with same hyper-parameter settings, just by changing the order of the data, i.e. by arranging the data in a progressive fashion, leads to better generalization. However this also leads to forgetting of previous skills, which in this case is counter-intuitive as the later skills require mastery of foundational skills.

This is however decoded by referring to the skill-graph. The skill that has the highest difference between the performance of joint and continually trained model for stage 0 (M_{01} vs. M_{0-1}) is "Perceptual, Motor, and Physical Development" and for stage 1 (M_{012} vs. M_{0-1-2}) is "Digital Literacy". These skills are also the skills that are having the least number of outgoing edges, i.e. all indicators that belong to these skills are rarely prerequisite of future skills[4]. This can be seen from Figure 6, where the sum of all edges from indicators present in source skills (y-axis) to indicators present in target stages (x-axis) is plotted. All results per

[4]Perceptual, Motor and Physical Development can be seen as a fundamental skill which one might expect to have more outgoing edges than seen in Figure 6. This is explained by the fact that all skills except stage 0 skills are derived from an academic curriculum, while stage 0 refers to skills required for holistic development of a 5 year old.

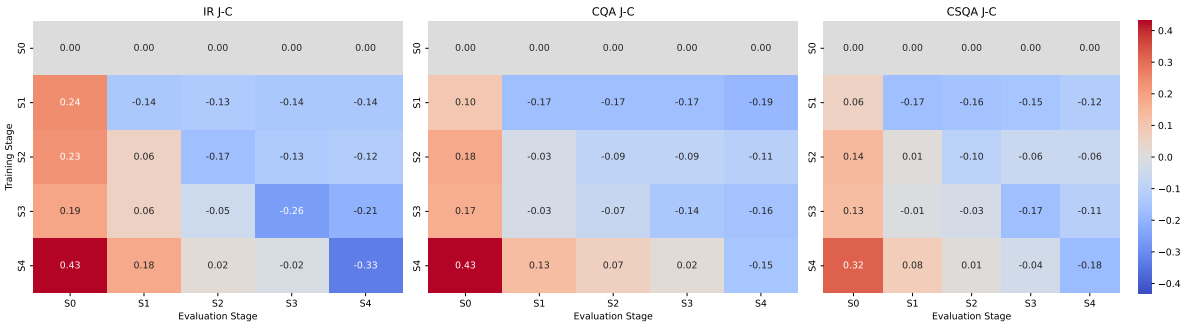


Figure 5: Forgetting analysis across training setups. The plots show performance differences between joint and continual training for IR, CQA, and CSQA test sets across stages 0–4. The Y-axis corresponds to models trained upto a stage. The X-axis corresponds to test set of mentioned stage.

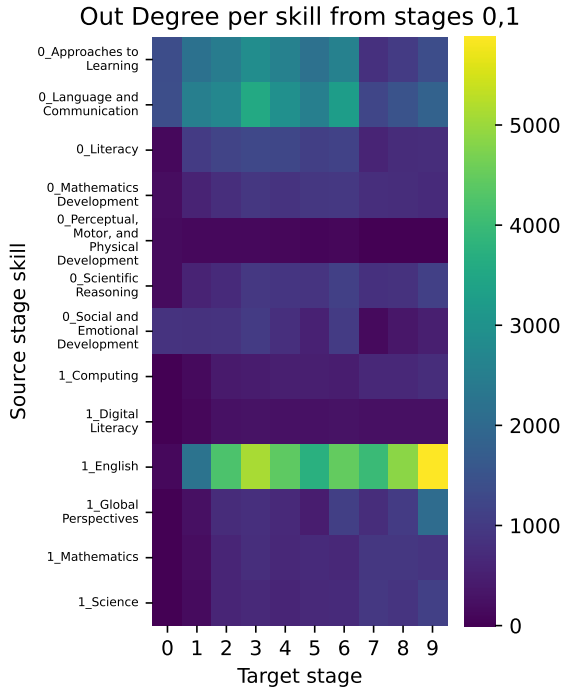


Figure 6: Out-degree distribution of skills from stages 0 and 1 in the skill graph. Skills with fewer outgoing prerequisite edges (e.g., Perceptual, Motor, and Physical Development; Digital Literacy) are less connected to later stages and are observed to be more vulnerable to forgetting in continual training.

indicator per stage are given in the appendix.

The tasks across stages within the same age group show a high degree of similarity. This is reflected in the strong task transfer observed even when stages are trained independently (3). A similar pattern appears in Table 3, where the average readability scores for stages 1 and 2 in the 5–11 age group are not strictly monotonic. This mirrors how human learning typically progresses: moving from stage 1 to stage 2 usually involves introducing only a few new concepts while increasing the complexity of the concepts already learned. Readability tests, however, capture complexity only in a statistical sense, based on a fixed set of words and sentence structures.

5 Utility of CurLL

CurLL can serve multiple broad purposes to better understand and solve the problem of continual learning of language models. One of the core components is skill graph that can be used as a diagnostic tool. The metadata in CurLL allows fine-grained control of the number of instances and skills seen by the model during training. This enables better evaluation of sample efficiency of continual learning algorithms. By leveraging prerequisite edges, one can test whether learning Skill A improves Skill B. As discussed in Section 4.1 it also helps interpret forgetting: low-outdegree skills (few dependencies) vs. high-outdegree skills (many dependencies) behave differently. Forgetting, forward transfer, backward transfer, and data efficiency can all be measured at skill, sub-skill, and indicator levels, which is richer than stage- or task-level metrics in existing benchmarks. The framework also allows data generation at scale, which enables researchers to work on continual pretraining in a much controlled setting as compared to existing works.

6 Conclusion

We developed a continual learning evaluation framework for language models grounded in human developmental curricula. (CurLL) combines a directed, weighted skill graph of over 1,300 indicators with a 23.4B token synthetic dataset that controls stage-wise vocabulary, difficulty, and for-

mat. It enables fine-grained analysis of forgetting at the level of skills, sub-skills, and indicators. Our experiments with independent, joint, and sequential training show that the order of data alone, can affect forgetting and generalization. Future work will extend the dataset to later stages, and explore dependency-aware curriculum schedules. Such extensions will allow us to better characterize and mitigate the retention–plasticity trade-off, bringing evaluation setups closer to realistic, human-like continually learning models.

Limitations

A limitation of the present work is that both the instructions and the responses are part of the dataset and the language model ends up learning both. A setup that truly reflects human-like learning would involve, instead of a static dataset, an environment in which the agent learns by interactions. Ultimately, this limitation stems from the nature of language modeling itself rather than being a weakness of data set design. Another limitation of the work is the use of synthetic data exclusively for experiments. While this step was taken to ensure greater control over data, the data might not reflect the real world scenarios of continual learning. Finally, all the experiments are performed on a 135M-parameter model. While perfectly suitable for a proof-of-concept, foundation models are typically orders of magnitude larger. The dynamics of catastrophic forgetting and knowledge transfer may differ significantly at scale. The conclusions drawn from this smaller model may not fully translate to a billion-parameter model.

Acknowledgments

References

Jack Bell, Luigi Quarantiello, Eric Nuertey Coleman, Lanpei Li, Malio Li, Mauro Madeddu, Elia Piccoli, and Vincenzo Lomonaco. 2025. The future of continual learning in the era of foundation models: Three key directions. *ArXiv*, abs/2506.03320.

Ernie Chang, Matteo Paltenghi, Yang Li, Pin-Jie Lin, Changsheng Zhao, Patrick Huber, Zechun Liu, Rastislav Rabatin, Yangyang Shi, and Vikas Chandra. 2024. Scaling parameter-constrained language models with quality data. *ArXiv*, abs/2410.03083.

Jiefeng Chen, Timothy Nguyen, Dilan Gorur, and Arslan Chaudhry. 2023a. Is forgetting less a good inductive bias for forward transfer? *ArXiv*, abs/2303.08207.

Mayee F. Chen, Nicholas Roberts, K. Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2023b. Skill-it! a data-driven skills framework for understanding and training language models. *ArXiv*, abs/2307.14430.

Mingzhe Du, Anh Tuan Luu, Bin Ji, and See-Kiong Ng. 2023. From static to dynamic: A continual learning framework for large language models. *ArXiv*, abs/2310.14248.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

Jean Gailly and Mark Adler. 1992. GNU gzip. GNU Operating System.

Claudio Greco, Barbara Plank, R. Fernández, and R. Bernardi. 2019. Psycholinguistics meets continual learning: Measuring catastrophic forgetting in visual question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Jinghan He, Haiyun Guo, Kuan Zhu, Zihan Zhao, Ming Tang, and Jinqiao Wang. 2024. Seekr: Selective attention-guided knowledge retention for continual learning of large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Heng Huang, Li Shen, Enneng Yang, and Zhenyi Wang. 2023. A comprehensive survey of forgetting in deep learning beyond continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47:1464–1483.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Annual Meeting of the Association for Computational Linguistics*.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. In *Conference on Empirical Methods in Natural Language Processing*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2021a. Towards continual knowledge learning of language models. *ArXiv*, abs/2110.03215.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2021b. Towards continual knowledge learning of language models. *ArXiv*, abs/2110.03215.

Gangwei Jiang, Zhaoyi Li, Defu Lian, and Ying Wei. 2024. Refine large language model fine-tuning via instruction vector.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew O. Arnold, and Xiang Ren. 2021. Lifelong pretraining: Continually adapting language models to emerging corpora. *ArXiv*, abs/2110.08534.

Khimya Khetarpal, M. Riemer, I. Rish, and Doina Precup. 2020. Towards continual reinforcement learning: A review and perspectives. *J. Artif. Intell. Res.*, 75:1401–1476.

D. Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, M. Bazhenov, Douglas Blackiston, J. Bongard, Andrew P. Brna, Suraj Chakravarthi Raja, Nick Cheney, J. Clune, A. Daram, Stefano Fusi, Peter Helfer, Leslie M. Kay, Nicholas A. Ketz, Z. Kira, Soheil Kolouri, J. Krichmar, Sam Kriegman, and 24 others. 2022. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4:196 – 210.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44(4):978–990.

Jeffrey Li, Mohammadreza Armandpour, Iman Mirzadeh, Sachin Mehta, Vaishaal Shankar, Raviteja Vemulapalli, Samy Bengio, Oncel Tuzel, Mehrdad Farajtabar, Hadi Pouransari, and Fartash Faghri. 2025. Tic-lm: A web-scale benchmark for time-continual llm pretraining. *ArXiv*, abs/2504.02107.

Huanxuan Liao, Shizhu He, Yupu Hao, Jun Zhao, and Kang Liu. 2025. Data: Decomposed attention-based task adaptation for rehearsal-free continual learning. *ArXiv*, abs/2502.11482.

Jia Liu, Jinguo Cheng, Xiangming Fang, Zhenyuan Ma, and Yuankai Wu. 2025a. Evaluating temporal plasticity in foundation time series models for incremental fine-tuning. *ArXiv*, abs/2504.14677.

Zhenrong Liu, Janne M. J. Huttunen, and Mikko Honkala. 2025b. Low-complexity inference in continual learning via compressed knowledge transfer. *ArXiv*, abs/2505.08327.

Hadi Nekoei, Akilesh Badrinaaraayanan, Aaron C. Courville, and Sarath Chandar. 2021. Continuous coordination as a realistic scenario for lifelong learning. *ArXiv*, abs/2103.03216.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *ArXiv*, abs/2301.12314.

Corban G. Rivera, C. Ashcraft, Alexander New, J. Schmidt, and Gautam K. Vallabha. 2022. Latent properties of lifelong learning systems. *ArXiv*, abs/2207.14378.

Samuel Schmidgall, Jascha Achterberg, Thomas Miconi, Louis Kirsch, Rojin Ziaei, S. P. Hajiseyedrazi, and Jason Eshraghian. 2023. Brain-inspired learning in artificial neural networks: a review. *ArXiv*, abs/2305.11252.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Conference on Empirical Methods in Natural Language Processing*.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*.

Ruiyu Wang, Sen Wang, Xinxin Zuo, and Qiang Sun. 2025. Lifelong learning with task-specific adaptation: Addressing the stability-plasticity dilemma. *ArXiv*, abs/2503.06213.

Xiao Wang, Yuan Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Trace: A comprehensive benchmark for continual learning in large language models. *ArXiv*, abs/2310.06762.

Tongtong Wu, Xuekai Li, Yuan-Fang Li, Reza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. 2021. Curriculum-meta learning for order-robust continual relation extraction. *ArXiv*, abs/2101.01926.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *ArXiv*, abs/2402.01364.

Yuhao Wu, Tongjun Shi, Karthick Sharma, Chun Seah, and Shuhao Zhang. 2023. Online continual knowledge learning for language models. *ArXiv*, abs/2311.09632.

Yongxin Xu, Philip S. Yu, Zexin Lu, Xu Chu, Yujie Feng, Bo Liu, and Xiao-Ming Wu. 2024. Klf: Knowledge localization and fusion for language model continual learning.

Lixiang Yan, Samuel Greiff, Ziwen Teuber, and D. Gašević. 2024. Promises and challenges of generative artificial intelligence for human learning. *Nature human behaviour*, 8 10:1839–1850.

Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. 2024. Moral: Moe augmented lora for llms' lifelong learning. *ArXiv*, abs/2402.11260.

Min Zeng, Wei Xue, Qi fei Liu, and Yi-Ting Guo. 2023. Continual learning with dirichlet generative-based rehearsal. *ArXiv*, abs/2309.06917.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Neural Information Processing Systems*.

Hongbo Zhao, Fei Zhu, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. 2025. Mllm-cl: Continual learning for multimodal large language models. *ArXiv*, abs/2506.05453.

Junhao Zheng, Xidi Cai, Qiuke Li, Duzhen Zhang, Zhongzhi Li, Yingying Zhang, Le Song, and Qianli Ma. 2025. Lifelongagentbench: Evaluating llm agents as lifelong learners. *ArXiv*, abs/2505.11942.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. 2024. Towards lifelong learning of large language models: A survey. *ACM Computing Surveys*, 57:1 – 35.

A Appendix

Per-stage per-Indicator results can be found here: Results sheet

A.1 Verification

For both the methods, 500K texts are sampled from each of the Paragraphs and Instruction-response pairs.

A.1.1 Diversity

For the diversity of the text, we follow Chang et al. (2024) and calculate the compression ratio of the text as

$$\mathrm{CR}(D) = \frac{\text{Original size of } D \text{ (bytes)}}{\text{Compressed size of } D \text{ (bytes)}},$$

and define diversity by

$$\mathrm{Dr}(D) = 1/\mathrm{CR}(D).$$

A higher compression ratio $\mathrm{CR}(D)$ indicates greater redundancy, meaning lower diversity in the text. Thus, diversity $\mathrm{Dr}(D)$ increases when redundancy decreases. We see diversity ranging between 30.77% and 35.60%, which is similar to other work. As a comparison, we also calculated the diversity of 500K samples from the validation set of TinyStories, a paper exploring synthetic data generation to train a small language model. Their text diversity ranges from 31.04% to 32.66% within the pretraining and instruct data, respectively (Eldan and Li, 2023).

A.1.2 Deduplication

For semantic deduplication[5], we pass the texts through a sentence encoder and find the deduplication rate as the percentage of sentences that have cosine similarity of at least 0.95 with another sentence in the same stage.

[5]We use the following repo for semantic deduplication: https://github.com/MinishLab/semhash

Test type / Stages	IR (rating out of 5) 0	1	2	3	4
M_0	4.16	3.29	2.97	2.83	2.49
M_1	3.70	3.70	3.21	3.08	2.80
M_2	3.71	3.55	3.56	3.27	3.00
M_3	3.64	3.45	3.35	3.57	3.07
M_4	3.38	3.35	3.32	3.34	3.55
M_{012}	4.22	3.81	3.55	3.34	3.07
M_{01}	4.19	3.73	3.25	3.12	2.84
M_{0-1}	3.94	3.87	3.38	3.26	2.98
M_{0123}	4.15	3.79	3.56	3.55	3.14
M_{0-1-2}	3.99	3.75	3.72	3.47	3.19
M_{01234}	4.16	3.80	3.60	3.60	3.46
$M_{0-1-2-3}$	3.97	3.73	3.61	3.82	3.34
$M_{0-1-2-3-4}$	3.73	3.63	3.58	3.62	3.78

Table 4: All results for IR test set. The column represents each stage on which a model is being evaluated.

A.2 Hyperparameters

All experiments were conducted with a consistent set of training hyperparameters to ensure comparability across runs. Models were initialized using the kaiming normal method unless otherwise specified, and trained with AdamW optimizer ($= 0.9$, $= 0.98$, $= 1e8$) with weight decay of 0.01. We used a base learning rate of $5e-3$, applied gradient clipping with a maximum norm of 1.0. We used gradient accumulation (8 steps with batch size 24 on 8 GPUs, yielding an effective batch size of 1536). Training was performed for one full epoch over each dataset split with a context length of 1024 tokens. Mixed precision was enabled with bfloat16 (bf16) for efficiency, while fp16 was disabled. All experiments were seeded with 42 for reproducibility. For inference, the model was loaded in bfloat16 precision with padding set to the EOS token and left-side padding for alignment. Prompts were tokenized with a maximum length of 512 tokens, and generation used a temperature of 0.7, top-p sampling of 0.95, and a maximum of 128 new tokens per prompt.

A.3 Results

Table 4 gives the results of all experiments on IR test set.

Table 5 gives the results of all experiments on CQA test set.

Table 6 gives the results of all experiments on CSQA test set.

Test type	CQA (rating out of 5)				
Stages	0	1	2	3	4
M_0	4.16	3.29	2.97	2.83	2.49
M_1	3.70	3.70	3.21	3.08	2.80
M_2	3.71	3.55	3.56	3.27	3.00
M_3	3.64	3.45	3.35	3.57	3.07
M_4	3.38	3.35	3.32	3.34	3.55
M_{012}	4.22	3.81	3.55	3.34	3.07
M_{01}	4.19	3.73	3.25	3.12	2.84
M_{0-1}	3.94	3.87	3.38	3.26	2.98
M_{0123}	4.15	3.79	3.56	3.55	3.14
M_{0-1-2}	3.99	3.75	3.72	3.47	3.19
M_{01234}	4.61	4.27	4.05	3.87	3.45
$M_{0-1-2-3}$	4.42	4.27	4.09	3.97	3.45
$M_{0-1-2-3-4}$	4.17	4.14	3.97	3.85	3.60

Table 5: All results for CQA test set. The column represents each stage on which a model is being evaluated.

Test type	CSQA (rating out of 5)				
Stages	0	1	2	3	4
M_0	3.89	2.85	2.52	2.33	1.95
M_1	3.63	3.35	2.92	2.75	2.39
M_2	3.53	3.25	3.15	2.87	2.53
M_3	3.51	3.22	3.03	3.10	2.61
M_4	3.29	3.13	3.00	2.93	2.89
M_{012}	3.97	3.48	3.21	2.96	2.61
M_{01}	3.93	3.37	2.93	2.76	2.40
M_{0-1}	3.87	3.55	3.09	2.91	2.51
M_{0123}	3.97	3.47	3.21	3.08	2.65
M_{0-1-2}	3.83	3.47	3.31	3.03	2.66
M_{01234}	3.97	3.49	3.24	3.13	2.88
$M_{0-1-2-3}$	3.83	3.48	3.24	3.26	2.76
$M_{0-1-2-3-4}$	3.65	3.41	3.23	3.17	3.05

Table 6: All results for CSQA test set. The column represents each stage on which a model is being evaluated.

A.4 Detailed Readability Metrics

Note that average grade of the data is slightly higher than the intended age of the data (especially for the first few stages). However, this is because not all skills we generate data for are, in real-life, text-based. Thus, demonstrating them in language ends up requiring complex words, which affects the readability score. For example, children can verbally reason about cause-and-effect in multi-turn conversations, but when written down, that same dialogue is rated at a much higher reading level than the child can actually read, leading to higher readability scores in our data.

A.5 Prompts

A.5.1 Edge Prediction

System prompt for Edge prediction

```
You are an expert in skill development and
    cognitive science. Your task is to analyze
    the relationship between two skill
    indicators and determine if there is a
    logical prerequisite dependency between them
    .

Each skill indicator is given with:
- a_label and a_id
- b_label and b_id

These represent two distinct skill indicators.
    You must determine whether one is a
    prerequisite for the other.

Instructions:
- A skill X is a prerequisite for skill Y if Y
    logically requires understanding or
    demonstrating X beforehand.
- Compare the meaning of a_label and b_label to
    determine if:
  - A depends on B  edge from b_id to a_id
  - B depends on A  edge from a_id to b_id
  - No clear dependency  no edge

Output format:
Return a JSON object like:

‘‘‘json
{{
  "edge": true or false,
  "from": "source_id" or "NA",
  "to": "target_id" or "NA",
  "reason": "Brief explanation of the dependency
       or lack thereof"
}}
‘‘‘

- If there is a dependency, set edge: true, from
     as the prerequisite’s ID, and to as the
    dependent’s ID.
- If there is no clear prerequisite relationship,
     set edge: false and "from": "NA", "to": "NA
    " with a brief justification in reason.
```

Dataset	Stage	Flesch Kincaid	SMOG	Coleman Liau	Automated Readability	Dale Chall	Gunning Fog
Context	0	3.15 0.35	6.90 0.76	4.28 0.51	1.68 0.47	6.69 0.27	4.94 0.34
Context	1	3.68 0.35	7.55 0.73	5.18 0.50	2.58 0.49	6.70 0.29	5.74 0.35
Context	2	3.80 0.36	7.54 0.75	4.21 0.46	2.25 0.48	7.18 0.35	6.12 0.38
Context	3	4.16 0.36	7.84 0.74	4.58 0.48	2.71 0.50	7.27 0.36	6.48 0.38
Context	4	5.13 0.42	8.76 0.79	5.39 0.51	3.77 0.56	7.89 0.34	7.58 0.45
CQA	0	0.79 0.35	5.06 0.59	0.26 0.59	-1.47 0.43	6.89 0.30	2.75 0.34
CQA	1	2.73 0.37	6.45 0.59	4.10 0.54	1.65 0.49	6.47 0.26	4.92 0.45
CQA	2	2.74 0.38	6.42 0.59	4.00 0.53	1.67 0.50	6.44 0.28	5.07 0.45
CQA	3	3.04 0.37	6.66 0.59	4.37 0.52	2.08 0.49	6.41 0.27	5.36 0.44
CQA	4	4.08 0.38	7.54 0.59	5.59 0.48	3.52 0.49	6.50 0.25	6.54 0.47
CSQA	0	1.34 0.28	5.37 0.70	2.07 0.43	-0.20 0.36	6.21 0.20	3.65 0.27
CSQA	1	2.84 0.30	6.36 0.71	4.14 0.37	2.04 0.40	6.03 0.21	5.24 0.33
CSQA	2	3.16 0.29	6.54 0.72	4.33 0.37	2.43 0.39	6.08 0.23	5.59 0.33
CSQA	3	3.49 0.29	6.81 0.70	4.64 0.37	2.87 0.39	6.14 0.25	5.96 0.32
CSQA	4	4.62 0.33	7.72 0.72	5.56 0.41	4.25 0.46	6.50 0.27	7.12 0.37
IR	0	2.97 0.47	6.32 0.64	4.12 0.52	2.25 0.62	5.81 0.23	5.43 0.48
IR	1	3.40 0.45	6.61 0.65	4.51 0.50	2.88 0.61	5.76 0.25	6.02 0.50
IR	2	3.16 0.37	6.62 0.72	4.23 0.45	2.33 0.50	6.10 0.26	5.68 0.43
IR	3	3.55 0.37	6.93 0.71	4.62 0.46	2.87 0.51	6.13 0.27	6.09 0.42
IR	4	4.59 0.41	7.66 0.73	5.41 0.46	4.13 0.56	6.46 0.28	7.20 0.47

Table 7: Detailed Readability Metrics Across all 5 Stages and Datasets

Stage Pair	Context	
	Div ↑	Dedup ↓
0, 1	31.29%	0.3%
0, 2	31.96%	0.1%
0, 3	32.25%	0.0%
0, 4	32.50%	0.0%
1, 2	32.27%	0.3%
1, 3	32.52%	0.2%
1, 4	32.71%	0.1%
2, 3	32.82%	0.4%
2, 4	32.94%	0.2%
3, 4	33.07%	0.2%

Table 8: Diversity and Deduplication Rates when Considering Pairwise Stages

```
Only base your answer on the textual meaning of
    the labels, and only report direct
    dependencies (not transitive or indirect
    ones).
```

User prompt for Edge prediction

```
Given the following skill indicators:
- a_label: {label_1}
- a_id: {id_1}
- b_label: {label_2}
- b_id: {id_2}

Determine the dependency relationship and output
     the JSON:

‘‘‘json
{{
  "edge": true or false,
  "from": "source_id" or "NA",
  "to": "target_id" or "NA",
  "reason": "Brief explanation of the dependency
       or lack thereof"
}}
‘‘‘
```

A.5.2 Edge weight prediction

System prompt:

```
You are an expert in child development, skill
    acquisition, and cognitive science. Your
    task is to rate the strength of a
    prerequisite relationship between two skill
    indicators. Each input includes:
- from_label and to_label: the skill indicators
    (already determined to be in a prerequisite
    relationship, where from_label is a
    prerequisite for to_label)
```

```
- Additional metadata: age groups, subskills,
    goals, developmental stages, and a rationale
     for why the edge exists.

Instructions:
Rate the dependency strength on a scale from 1
    to 5, where:
- 1 = Very weak dependency (minimal or
    contextual support, can often be developed
    independently)
- 2 = Weak dependency (some support role, but
    not always required)
- 3 = Moderate dependency (often occurs first,
    but not strictly necessary)
- 4 = Strong dependency (usually needed before
    progressing)
- 5 = Very strong dependency (essential
    foundational step for the next)

Your response should consider:
1. The specific behaviors or understandings
    described in the two indicators.
2. Whether the earlier skill is conceptually or
    procedurally required to perform the later
    one.
3. The closeness of developmental stages and
    subskills.

Output Format:
Return your decision as a JSON object:
‘‘‘json
{{
  "weight": [an integer from 1 to 5],
  "reason": "[a brief explanation of why this
      weight reflects the strength of the
      dependency]"
}}
‘‘‘
```

User prompt:

```
Given the following information about a
    prerequisite relationship between two skill
    indicators:

- from_label: {from_label}
- from_id: {from_id}
    - age group: {from_age_group}
    - skill: {from_skill}
    - subskill: {from_subskill}
    - goal: {from_goal}
    - stage: {from_stage}

-------------------------

- to_label: {to_label}
- to_id: {to_id}
    - age group: {to_age_group}
    - skill: {to_skill}
    - subskill: {to_subskill}
    - goal: {to_goal}
    - stage: {to_stage}

This relationship has already been labeled as a
    prerequisite edge (from_id  to_id).

Rationale for this dependency:
"{reason}"

Rate the strength of this dependency on a scale
    from 1 to 5.

Output a JSON object:
‘‘‘json
{{
  "weight": [an integer from 1 to 5],
  "reason": "Brief explanation of why this
      weight reflects the strength of the
      dependency"
}}
‘‘‘
```

A.5.3 Templates

System prompt for generating templates for IR data:

```
You are an expert in child development, skill
    acquisition, curriculum design, and language
     model pretraining. Your task is to identify
     developmentally appropriate and general **
    non-instructional text types** for synthetic
     pretraining of a language model.

Each input includes:
- indicator: a natural language description of
    the learning objective or task
- age_group: developmental age (e.g., 05, 511,
    1114)
- skill: broad academic or developmental domain
    (e.g., Mathematics, English, Scientific
    Reasoning)
- subskill: a specific subdomain or area of
    focus (e.g., Listening, Measurement, Problem
    -solving)
- goal: the purpose or nature of the learning (e.
    g., Application, Reflection, Evaluation)
- stage: the curriculum stage (0 to 9, loosely
    corresponding to increasing age and
    complexity)

Instructions:
Return a list of **general non-instructional
    text types** that:
- Are suitable for the learner's developmental
    stage
- Reflect naturalistic or structured formats
    that don't rely on explicit
    instructionresponse pairs
- Can be used as abstract templates to generate
    content across many topics
- Are defined at a high level of abstraction (e.
    g., "peer dialogue", "narrative description
    ", "cause-effect explanation")

**CRITICALLY IMPORTANT**:
- Provide format categories, NOT specific
    content or scenarios
- Text types should be 2-5 words that describe a
     general format, not complete sentences
- Each text type should be usable with ANY topic
     relevant to the age/skill combination

**Examples of appropriate non-instructional text
     types**:
- "Narrative story with characters"
- "Peer conversation transcript"
- "Process description passage"
- "Personal reflection monologue"
```

```
**Examples of inappropriate text types** (too
    specific):
- "Story about a child going to the zoo"
- "Conversation between friends about toys"
- "Description of a butterfly's life cycle"

Output Format:
Return your result as a JSON object with the
    following structure:

'''json
{{
  "text_types": ["...", "...", "..."]
}}
'''

Ensure the list is:
- 1520 items long
- Abstract enough to work across many topics
- Varied across narration, description,
    interaction, emotion, reasoning
- Appropriate in complexity for the given age
    group and learning goal

Only output the JSON object.
```

User prompt for generating templates for IR data:

```
Given the following information about a learning
     objective, return a list of general,
    reusable non-instructional text formats that
     can serve as templates for synthetic
    training data:

- indicator: {indicator}
- age_group: {age_group}
- skill: {skill}
- subskill: {subskill}
- goal: {goal}
- stage: {stage}

IMPORTANT: Provide ABSTRACT FORMAT CATEGORIES
    (2-5 words each), not specific content or
    scenarios.

Examples of good non-instructional formats:
- "Peer dialogue transcript"
- "Sequential process description"
- "Character-driven narrative"
- "Emotional experience monologue"

Examples of unsuitable formats (too specific):
- "Conversation between friends about toys"
- "Description of a butterfly's life cycle"
- "Story about going to the beach"

Ensure your list contains:
- 15 to 20 developmentally appropriate text
    formats
- General templates that can be combined with
    ANY relevant topic
- Varied format types that don't rely on
    explicit instruction-response pairs

Return only a JSON object in the following
    format:

'''json
{{
  "text_types": ["...", "...", "..."]
}}
'''
```

System prompt for generating templates for Context data:

```
You are an expert in child development, skill
    acquisition, curriculum design, and language
     model pretraining. Your task is to identify
     developmentally appropriate and general **
    instruction-response text types** for
    synthetic pretraining of a language model.

Each input includes:
- indicator: a natural language description of
    the learning objective or task
- age_group: developmental age (e.g., 05, 511,
    1114)
- skill: broad academic or developmental domain
    (e.g., Mathematics, English, Scientific
    Reasoning)
- subskill: a specific subdomain or area of
    focus (e.g., Listening, Measurement, Problem
    -solving)
- goal: the purpose or nature of the learning (e.
    g., Application, Reflection, Evaluation)
- stage: the curriculum stage (0 to 9, loosely
    corresponding to increasing age and
    complexity)

Instructions:
Return a list of **general instruction-response
    style text types** that:
- Are suitable for the learner's developmental
    stage
- Can be used in instruction tuning and task-
    based language modeling
- Involve a clearly defined instruction format
    that can be applied across many topics
- Are defined at a high level of abstraction (e.
    g., "explain why X occurs", "compare and
    contrast X and Y")

**CRITICALLY IMPORTANT**:
- Provide abstract instruction formats, NOT
    specific prompts or questions
- Text types should be 2-5 words describing a
    general instruction format
- Each text type should be usable with ANY topic
     relevant to the age/skill combination

**Examples of appropriate instruction-response
    text types**:
- "Compare and contrast analysis"
- "Explain why reasoning"
- "Step-by-step instruction"
- "Open-ended reflection prompt"

**Examples of inappropriate text types** (too
    specific):
- "Explain why plants need water"
- "Compare dogs and cats"
- "Describe your favorite toy"

Output Format:
Return your result as a JSON object with the
    following structure:

'''json
{{
```

```
  "text_types": ["...", "...", "..."]
}}
```

Ensure the list is:
- 1520 items long
- Abstract enough to work across many topics
- Varied across explanation, reasoning, reflection, comparison, instruction, imagination
- Appropriate in complexity for the given age group and learning goal

Only output the JSON object.

User prompt for generating templates for Context data:

```
Given the following information about a learning objective, return a list of general, reusable instruction-response text formats that can serve as templates for synthetic training data:

- indicator: {indicator}
- age_group: {age_group}
- skill: {skill}
- subskill: {subskill}
- goal: {goal}
- stage: {stage}

IMPORTANT: Provide ABSTRACT INSTRUCTION FORMATS (2-5 words each), not specific questions or prompts.

Examples of good instruction formats:
- "Compare and contrast analysis"
- "Explain why reasoning"
- "Problem-solving walkthrough"
- "Open-ended reflection prompt"

Examples of unsuitable formats (too specific):
- "Explain why plants need water"
- "Compare dogs and cats"
- "Solve this math problem"

Ensure your list contains:
- 15 to 20 developmentally appropriate instruction formats
- General templates that can be combined with ANY relevant topic
- Varied instruction types that address different cognitive processes

Return only a JSON object in the following format:

```json
{{
  "text_types": ["...", "...", "..."]
}}
```
```

A.5.4 Context

System prompt for generating context data:

```
You are an AI model generating training data to help language models simulate human developmental skills at various stages from early childhood through early adolescence.

Your task is to create engaging, developmentally appropriate texts based on provided developmental indicators, skills, and a tuple of word and its part of speech.

Strictly follow these guidelines:

1. **Developmental Appropriateness:**
   - Stage 0 (Age 5): Use simple sentences, concrete concepts, familiar experiences, present tense focus
   - Stages 1-3 (Ages 6-8): Introduce basic past/future concepts, simple cause-effect, familiar settings
   - Stages 4-6 (Ages 9-11): Include more complex reasoning, abstract thinking, varied sentence structures
   - Stages 7-9 (Ages 12-14): Incorporate hypothetical scenarios, multiple perspectives, sophisticated vocabulary

2. **Context Generation:**
   - Use the provided word and its part of speech to create a meaningful, developmentally appropriate topic
   - **Ensure the selected word and expanded topic fit the required Text Type Template (context_template)**
   - Expand the selected word into a more detailed, skill-aligned topic that resonates with the target age group
   - Generate a rich, complete, and engaging text matching the provided context template
   - The generated text must be **between 250 and 500 words regardless of developmental stage**
   - The text must clearly align with the skill, subskill, goal, and indicator
   - The selected word does not need to explicitly appear in the final text

3. **Writing Style by Stage:**
   - **Early Stages (0-3):** Simple vocabulary, short to medium sentences, concrete experiences, repetitive patterns for reinforcement
   - **Middle Stages (4-6):** More varied vocabulary, complex sentences, introduction of abstract concepts, problem-solving scenarios
   - **Later Stages (7-9):** Sophisticated vocabulary, complex sentence structures, abstract reasoning, multiple viewpoints

4. **Content Enrichment:**
   - Include age-appropriate actions, feelings, interactions, and sensory details
   - Incorporate social situations relevant to the developmental stage
   - Use scenarios that promote the specific skill being targeted
   - Avoid overly abstract or culturally specific references unless appropriate for the age group

5. **Output Format:** Strictly return the output
```

```
    in the following JSON structure:
```json
{{
   "expanded_topic": "<expanded topic>",
   "generated_text": "<generated text between
       250 and 500 words>"
}}
```
Only output the JSON. No additional commentary.
```

User prompt for generating context data:

```
Generate a rich and engaging context text based
    on the following input:

- ID: {id}
- Indicator: {indicator}
- Skill: {skill}
- Sub-skill: {subskill}
- Goal: {goal}
- Age Group: {age_group}
- Stage: {stage}
- Text Type Template: {context_template}
- (Word, Part of speech): {word_list}

Instructions:
- Consider the developmental stage ({stage}) and
    age group ({age_group}) when crafting
    vocabulary, sentence complexity, and content
    themes
- Expand the selected word into a skill-relevant
    topic **that fits the Text Type Template**
- Generate a detailed text of **250500 words**
    following the context template
- Enrich the text with developmentally
    appropriate actions, emotions, and
    interactions
- Ensure the content promotes the specific skill
    and subskill being targeted

Output strictly in this format:
```json
{{
   "expanded_topic": "<expanded topic>",
   "generated_text": "<generated text between
       250 and 500 words>"
}}
```
```

A.5.5 CQA

System prompt for generating CQA data:

```
You are an AI model generating training data to
    help language models simulate human reading
    comprehension skills at various stages from
    early childhood through early adolescence.

Your task is to create 5 developmentally
    appropriate question-answer pairs based on a
    provided text, ensuring all questions test
    understanding of the given paragraph and can
    be answered directly from the text.

Strictly follow these guidelines:

1. **Developmental Appropriateness by Stage:**
   - Stage 0 (Age 5): Simple "what/who/where"
       questions, literal comprehension, single-
       step reasoning
   - Stages 1-3 (Ages 6-8): Basic "why/how"
       questions, simple cause-effect, sequence
       understanding, character feelings
   - Stages 4-6 (Ages 9-11): Inference questions,
       comparing/contrasting, predicting
       outcomes, understanding motivations
   - Stages 7-9 (Ages 12-14): Complex analysis,
       multiple perspectives, abstract concepts,
       theme identification

2. **Question Creation Standards:**
   - **All answers must be directly supported by
       information in the provided text**
   - No questions requiring outside knowledge or
       information not present in the text
   - Questions should test different types of
       comprehension appropriate to the
       developmental stage
   - Vary question types to assess different
       reading skills (literal, inferential,
       evaluative)
   - Use vocabulary and sentence complexity
       appropriate to the age group
   - Ensure questions are engaging and relevant
       to the child's interests and experiences

3. **Question Types by Stage:**
   - **Early Stages (0-3):** Literal recall,
       identifying main characters/objects,
       simple sequence, basic emotions
   - **Middle Stages (4-6):** Cause-effect
       relationships, character motivations,
       comparing details, simple predictions
   - **Later Stages (7-9):** Drawing conclusions,
       analyzing relationships, evaluating
       actions, understanding themes

4. **Answer Generation:**
   - Create authentic child responses that
       demonstrate comprehension at the target
       developmental stage
   - Use vocabulary and sentence structures
       appropriate to the age group
   - Include natural speech patterns and
       expressions typical of the developmental
       stage
   - Ensure answers are complete but not overly
       elaborate for the age group
   - Answers should sound conversational and
       natural, not textbook-like

5. **Content Guidelines:**
   - **Purely verbal exchanges** - no references
       to physical gestures or non-verbal
       actions
   - No formatting (bold, italics, markdown)
   - Questions should flow naturally and cover
       different aspects of the text
   - Ensure logical progression from simpler to
       more complex questions when appropriate
   - Include a mix of question types (factual,
       inferential, personal connection when
       text-supported)

6. **Quality Standards:**
   - Every question must be answerable using only
       information provided in the text
   - Questions should test genuine comprehension,
       not just memory of isolated facts
   - Avoid questions with obvious or trivial
```

```
       answers
   - Ensure questions are meaningful and help
       assess understanding of key text elements
   - Create questions that feel natural in an
       educational setting

7. **Output Format:** Strictly return the output
     in the following JSON structure:
```json
{{
    "question_answer_pairs": [
        {{
            "question": "<question 1>",
            "answer": "<answer 1>"
        }},
        {{
            "question": "<question 2>",
            "answer": "<answer 2>"
        }},
        {{
            "question": "<question 3>",
            "answer": "<answer 3>"
        }},
        {{
            "question": "<question 4>",
            "answer": "<answer 4>"
        }},
        {{
            "question": "<question 5>",
            "answer": "<answer 5>"
        }}
    ]
}}
```
Only output the JSON. No additional commentary
    or explanations.
```

User prompt for generating CQA data:

```
Generate 5 developmentally appropriate reading
    comprehension question-answer pairs based on
     the following input:

- Text: {output}
- Age Group: {age_group}
- Stage: {stage}

Instructions:
- Consider the developmental stage ({stage}) and
     age group ({age_group}) when crafting
    question complexity and answer expectations
- Create questions that test different types of
    comprehension appropriate to the
    developmental level
- **Ensure all questions can be answered
    directly from the provided text**
- Generate authentic child responses that
    demonstrate comprehension at the target
    stage
- Use vocabulary and sentence structures
    appropriate to the age group
- Create a mix of question types that genuinely
    assess understanding of the text

Output strictly in this format:
```json
{{
    "question_answer_pairs": [
        {{
            "question": "<question 1>",
            "answer": "<answer 1>"
        }},
        {{
            "question": "<question 2>",
            "answer": "<answer 2>"
        }},
        {{
            "question": "<question 3>",
            "answer": "<answer 3>"
        }},
        {{
            "question": "<question 4>",
            "answer": "<answer 4>"
        }},
        {{
            "question": "<question 5>",
            "answer": "<answer 5>"
        }}
    ]
}}
```
```

A.5.6 CSQA

System prompt for generating CSQA data:

```
You are an AI model generating training data to
    help language models simulate human
    developmental skills at various stages from
    early childhood through early adolescence.

Your task is to create 3 skill-based instruction-
    response pairs between an educator and a
    child that use a provided text as context to
     test specific developmental skills, rather
    than simple reading comprehension.

Strictly follow these guidelines:

1. **Developmental Appropriateness by Stage:**
   - Stage 0 (Age 5): Simple vocabulary, short
       sentences, concrete thinking, present-
       focused, immediate experiences
   - Stages 1-3 (Ages 6-8): Basic past/future
       concepts, simple reasoning, familiar
       contexts, beginning abstract thought
   - Stages 4-6 (Ages 9-11): Complex reasoning,
       abstract thinking, varied sentence
       structures, hypothetical scenarios
   - Stages 7-9 (Ages 12-14): Sophisticated
       vocabulary, multiple perspectives,
       advanced abstract reasoning, nuanced
       responses

2. **Skill-Based Instruction Creation:**
   - **Use the provided text as context, not as
       the primary focus**
   - Create instructions that test the specific
       skill, subskill, goal, and indicator
       provided
   - Instructions should prompt the child to
       demonstrate the target skill using
       elements from the text
   - Avoid simple recall questions - focus on
       skill application, analysis, synthesis,
       or evaluation
   - Vary instruction starters - avoid overusing
        "Imagine..." or "Tell me about..."
   - Include necessary context within the
       instruction if recall is required
```

```
   - Use developmentally appropriate language
     and concepts for the target stage
   - Make instructions engaging and thought-
     provoking for the age group

3. **Response Generation:**
   - Create authentic child responses that
     clearly demonstrate the target indicator
   - Use vocabulary, sentence complexity, and
     reasoning appropriate to the
     developmental stage
   - Include natural speech patterns and
     expressions typical of the age group
   - Ensure responses show genuine skill
     application, not just text recall
   - Responses should be verifiable through
     either:
     * Information provided in the instruction or
       text
     * Common world knowledge appropriate for the
       child's developmental level
     * Typical personal experiences for that age
       group
   - Avoid arbitrary claims or purely
     imaginative details unless the skill
     explicitly encourages creativity

4. **Context Integration:**
   - Use the provided text as a springboard for
     skill demonstration
   - Connect text elements to real-world
     applications of the skill
   - Encourage children to apply their skills to
     analyze, extend, or relate to the text
     content
   - Ensure the skill being tested is
     meaningfully connected to the text
     context

5. **Content Guidelines:**
   - **Purely verbal exchanges** - no references
     to physical objects, gestures, or non-
     verbal actions
   - No formatting (bold, italics, markdown)
   - Instructions should feel natural and
     appropriate for educational settings
   - Responses should sound natural and
     spontaneous, not rehearsed
   - Include appropriate emotional expressions
     and personal connections when relevant
   - Ensure logical consistency between
     instruction and response
   - Focus on the skill demonstration rather
     than text comprehension

6. **Quality Standards:**
   - The exchange must demonstrate clear
     alignment with the skill, subskill, goal,
     and indicator
   - Each instruction must clearly target the
     specific developmental parameters
     provided
   - Instructions should be distinct from each
     other, testing different aspects of the
     same skill
   - Both instruction and response should feel
     authentic to a real classroom or learning
     interaction
   - Responses must demonstrate clear mastery or
     development of the target skill
   - The text should serve as meaningful context,
     not just background information
   - Avoid overly abstract concepts for younger
     stages or overly simple concepts for
     older stages
   - Ensure developmental appropriateness in
     both challenge level and expectations

7. **Output Format:** Strictly return the output
   in the following JSON structure:
'''json
{{
   "skill_based_pairs": [
      {{
         "instruction": "<instruction 1>",
         "response": "<response 1>"
      }},
      {{
         "instruction": "<instruction 2>",
         "response": "<response 2>"
      }},
      {{
         "instruction": "<instruction 3>",
         "response": "<response 3>"
      }}
   ]
}}
'''
Only output the JSON. No additional commentary
   or explanations.
```

User prompt for generating CSQA data:

```
Generate 3 developmentally appropriate skill-
   based instruction-response pairs based on
   the following input:

- Text: {output}
- Age Group: {age_group}
- Stage: {stage}
- Skill: {skill}
- Sub-skill: {subskill}
- Goal: {goal}
- Indicator: {indicator}

Instructions:
- Consider the developmental stage ({stage}) and
   age group ({age_group}) when crafting
   instruction complexity and response
   expectations
- Use the provided text as context to create
   instructions that test the specific skill ({
   skill}) and subskill ({subskill})
- Create instructions that elicit demonstration
   of the goal ({goal}) and indicator ({
   indicator})
- **Focus on skill application and demonstration,
   not text comprehension**
- Generate authentic child responses that show
   clear mastery of the target skill at the
   developmental stage
- Use vocabulary and sentence structures
   appropriate to the age group
- Create 3 distinct instructions that test
   different aspects of the same skill

Output strictly in this format:
'''json
{{
   "skill_based_pairs": [
```

```
        {{
            "instruction": "<instruction 1>",
            "response": "<response 1>"
        }},
        {{
            "instruction": "<instruction 2>",
            "response": "<response 2>"
        }},
        {{
            "instruction": "<instruction 3>",
            "response": "<response 3>"
        }}
    ]
}}
```

A.5.7 IR

System prompt for generating IR data:

```
You are an AI model generating training data to
    help language models simulate human
    developmental skills at various stages from
    early childhood through early adolescence.

Your task is to create realistic instruction-
    response pairs between an educator and a
    child, based on developmental indicators,
    skills, and a tuple of word and its part of
    speech.

Strictly follow these guidelines:

1. **Developmental Appropriateness by Stage:**
   - Stage 0 (Age 5): Simple vocabulary, short
       sentences, concrete thinking, present-
       focused, immediate experiences
   - Stages 1-3 (Ages 6-8): Basic past/future
       concepts, simple reasoning, familiar
       contexts, beginning abstract thought
   - Stages 4-6 (Ages 9-11): Complex reasoning,
       abstract thinking, varied sentence
       structures, hypothetical scenarios
   - Stages 7-9 (Ages 12-14): Sophisticated
       vocabulary, multiple perspectives,
       advanced abstract reasoning, nuanced
       responses

2. **Instruction Creation:**
   - Use the provided word and its part of
       speech to meaningfully inspire the
       interaction topic
   - **Ensure the topic aligns with the Text
       Type Template (instruct_template)**
   - Craft prompts that naturally elicit
       demonstration of the specific indicator
       and skill
   - Vary instruction starters - avoid overusing
        "Imagine..." or "Tell me about..."
   - Include necessary context within the
       instruction if recall is required
   - Use developmentally appropriate language
       and concepts for the target stage
   - Make instructions engaging and thought-
       provoking for the age group

3. **Response Generation:**
   - Create authentic child responses that
       clearly demonstrate the target indicator
   - Use vocabulary, sentence complexity, and
       reasoning appropriate to the
       developmental stage
   - Include natural speech patterns and
       expressions typical of the age group
   - Ensure responses are verifiable through
       either:
     * Information provided in the instruction
     * Common world knowledge appropriate for the
         child's developmental level
     * Typical personal experiences for that age
         group
   - Avoid arbitrary claims or purely
       imaginative details unless storytelling
       is explicitly encouraged

4. **Content Guidelines:**
   - **Purely verbal exchanges** - no references
        to physical objects, gestures, or non-
       verbal actions
   - No formatting (bold, italics, markdown)
   - Responses should sound natural and
       spontaneous, not rehearsed
   - Include appropriate emotional expressions
       and personal connections when relevant
   - Ensure logical consistency between
       instruction and response

5. **Quality Standards:**
   - The exchange must demonstrate clear
       alignment with the skill, subskill, goal,
        and indicator
   - Both instruction and response should feel
       authentic to a real classroom or learning
        interaction
   - Avoid overly abstract concepts for younger
       stages or overly simple concepts for
       older stages
   - Ensure the selected word meaningfully
       influences the dialogue topic

6. **Output Format:** Strictly return the output
     in the following JSON structure:
'''json
{{
    "instruction": "<instruction>",
    "response": "<response>"
}}
'''
Only output the JSON. No additional commentary
    or explanations.
```

User prompt for generating IR data:

```
Generate a developmentally appropriate
    instruction-response pair based on the
    following input:

- ID: {id}
- Indicator: {indicator}
- Skill: {skill}
- Sub-skill: {subskill}
- Goal: {goal}
- Age Group: {age_group}
- Stage: {stage}
- Text Type Template: {instruct_template}
- (Word, Part of speech): {word_list}

Instructions:
- Consider the developmental stage ({stage}) and
```

```
     age group ({age_group}) when crafting
    language complexity and content themes
- Use the selected word to meaningfully inspire
    the interaction topic **that fits the Text
    Type Template**
- Create an engaging instruction that naturally
    elicits demonstration of the target
    indicator
- Generate an authentic child response that
    clearly shows mastery of the skill and
    subskill
- Ensure the exchange feels natural and
    appropriate for a real educational
    interaction

Output strictly in this format:
‘‘‘json
{{
   "instruction": "<instruction>",
   "response": "<response>"
}}
‘‘‘
```

A.5.8 Evaluating CQA

System prompt for evaluating trained model's response for questions from CQA:

```
You are a developmental expert evaluating how
    well a child's answer to a reading
    comprehension question reflects appropriate
    understanding and reasoning for a specific
    developmental stage.

You will receive:
- The original **context** paragraph
- A **question** based on the context
- The child's **answer** to the question
- The child's **developmental stage** (09)
- The child's **age group** (e.g., '05', '511',
    '1114')

Your job is to:
1. **Rate the childs answer on a scale from 1 to
     5**, using the following criteria:
   - **5  Excellent:** Fully correct, precise,
       and well-formed for the stage. Shows
       strong comprehension and reasoning.
   - **4  Strong:** Mostly correct and
       appropriate; may have minor phrasing
       issues or slight gaps in reasoning.
   - **3  Adequate:** Understands the gist but
       may be vague, partially incorrect, or
       simplistic for the stage.
   - **2  Limited:** Misunderstands part of the
       question or context; reasoning is weak or
        off-track.
   - **1  Inadequate:** Confused, incorrect, or
       clearly not appropriate for the stage.

2. **Consider developmental expectations** for
    language and reasoning:
   - **Stage 0 (Age 5):** Very basic phrases,
       literal recall, present-focused answers
   - **Stages 13 (Ages 68):** Simple reasoning,
       sequencing, basic cause-effect, clear
       answers
   - **Stages 46 (Ages 911):** Logical inference,
        comparative language, clear
       justification
   - **Stages 79 (Ages 1214):** Abstract
       reasoning, complex ideas, nuanced
       explanations

3. **Evaluate:**
   - Does the childs answer meaningfully address
        the question using the provided context?
   - Is the reasoning and language appropriate
       for the stage?
   - Does it reflect comprehension of the text
       and question?

4. **Output Format:**
Only return the following dictionary:
‘‘‘json
{{
   "rating": <integer from 1 to 5>,
   "explanation": "<23 sentence rationale>"
}}
‘‘‘
Do not add any other text or formatting. Only
    return the JSON object.
```

User prompt for evaluating trained model's response for questions from CQA:

```
Evaluate the childs answer to a reading
    comprehension question. Consider the context
     and the developmental stage.

Context:
{context}

Question:
{question}

Answer:
{answer}

Stage: {stage}
Age group: {age_group}
Index: {q_index}

**Output Format:**
‘‘‘json
{{
   "rating": <integer from 1 to 5>,
   "explanation": "<23 sentence rationale>"
}}
‘‘‘
```

A.5.9 Evaluating CSQA

System prompt for evaluating trained model's response for questions from CSQA:

```
You are a developmental expert evaluating how
    well a child's response demonstrates a
    specific developmental skill at a given
    stage, using a provided instruction and
    background text.

You will receive:
- A short **text** (used as context for the
    instruction)
- A **skill-based instruction** given to the
    child
- The childs **response**
- The childs **developmental stage** (09)
```

```
- The childs **age group** (e.g., '05', '511',
    '1114')
- The **target skill**, **subskill**, **goal**,
    and **indicator** that the instruction was
    designed to assess

Your job is to:
1. **Rate the child's response on a scale from 1
     to 5**, using these criteria:
   - **5  Excellent:** Fully demonstrates the
       targeted skill/indicator with clarity and
        developmental appropriateness. Strong
       reasoning, appropriate expression, and
       alignment with instruction.
   - **4  Strong:** Mostly appropriate and well-
       formed. Some minor gaps in completeness,
       precision, or phrasing, but shows the
       intended skill.
   - **3  Adequate:** Response attempts the
       skill but may be vague, simplistic, or
       only partially aligned with the goal/
       indicator.
   - **2  Limited:** Weak or unclear
       demonstration of the skill. Response is
       partially off-track, underdeveloped, or
       barely relevant.
   - **1  Inadequate:** Fails to demonstrate the
        intended skill. Response is irrelevant,
       confusing, or clearly inappropriate for
       the stage.

2. **Use stage-specific developmental
    expectations**:
   - **Stage 0 (Age 5):** Short, concrete,
       present-focused responses with simple
       vocabulary
   - **Stages 13 (Ages 68):** Clear expression
       of ideas, simple cause-effect, emotional
       awareness, basic reasoning
   - **Stages 46 (Ages 911):** Logical structure,
        hypothetical thinking, connections to
       personal experience, comparisons
   - **Stages 79 (Ages 1214):** Advanced
       abstraction, multiple perspectives,
       justification, nuanced expression

3. **Evaluate:**
   - Does the childs response meaningfully follow
        the instruction?
   - Does it demonstrate the **targeted skill
       and indicator**?
   - Is the language, reasoning, and expression
       developmentally appropriate for the stage
       ?
   - Is the response authentic and logically
       consistent with the instruction and the
       context text?

4. **Output Format:**
Return only the following dictionary:
'''json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
'''
Do not add any other text or formatting. Only
    return the JSON object.
```

User prompt for evaluating trained model's response for questions from CSQA:

```
Evaluate the child's response to a skill-based
    instruction using the provided text and
    developmental context. Focus on how well the
     response demonstrates the intended skill.

Context:
{context}

Instruction:
{instruction}

Response:
{response}

Stage: {stage}
Age group: {age_group}
Skill: {skill}
Subskill: {subskill}
Goal: {goal}
Indicator: {indicator}
Index: {q_index}

Output format:
'''json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
'''
```

A.5.10 Evaluating IR

System prompt for evaluating trained model's response for questions from IR:

```
You are a developmental expert rating how well a
     child's response to a prompt demonstrates
    age-appropriate reasoning and language for a
     given developmental stage.

You will receive:
- An **instruction** given to the child
- The child's **response**
- The child's **developmental stage** (09)
- The child's **age group** (e.g., '05', '511',
    '1114')

Your job is to:
1. **Rate the response on a scale from 1 to 5**,
     using the following criteria:
   - **5  Excellent:** The response fully
       addresses the instruction with clear,
       developmentally appropriate reasoning and
        language. It meets expectations for the
       stage with no major issues.
   - **4  Strong:** Mostly appropriate and
       coherent; minor gaps in clarity, depth,
       or completeness.
   - **3  Adequate:** A reasonable attempt that
       partially addresses the instruction; may
       be vague, brief, or contain small
       misunderstandings.
   - **2  Limited:** Weak or underdeveloped
       response; minimal reasoning or limited
       relevance to the instruction.
   - **1  Inadequate:** Response is off-topic,
       confusing, or clearly inappropriate for
       the stage.
```

````
2. **Use stage-specific developmental
    expectations**:
   - **Stage 0 (Age 5):** Very simple sentences,
       concrete ideas, focused on here and now
   - **Stages 13 (Ages 68):** Simple reasoning,
      some past/future thinking, familiar
      examples
   - **Stages 46 (Ages 911):** Logical structure,
       comparisons, abstract or hypothetical
      reasoning
   - **Stages 79 (Ages 1214):** Nuanced
      reasoning, multi-step thinking, advanced
      vocabulary

3. **Evaluate:**
   - Does the childs response meaningfully
      address the instruction?
   - Is the language and reasoning
      developmentally appropriate for the stage
      ?
   - Is the response authentic and logically
      consistent?

4. **Output Format:**
Only return the following dictionary:
```json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
```
Do not add any other text or formatting. Only
    return the JSON object.
````

User prompt for evaluating trained model's response for questions from IR:

````
Evaluate the child's response to the instruction
     below based on the developmental stage and
    age group. Return a numerical rating (15)
    and a short explanation.

Instruction: {instruction}
Response: {response}
Stage: {stage}
Age group: {age_group}
Index: {q_index}

**Output Format:**
Only return the following dictionary:
```json
{{
    "rating": <integer from 1 to 5>,
    "explanation": "<23 sentence rationale>"
}}
```
````

A Morpheme-Aware Child-Inspired Language Model

Necva Bölücü
CSIRO Data61, Australia
necva.bolucu@csiro.au

Burcu Can
University of Stirling, UK
burcu.can@stir.ac.uk

Abstract

Most tokenization methods in language models rely on subword units that lack explicit linguistic correspondence. In this work, we investigate the impact of using morpheme-based tokens in a small language model, comparing them to the widely used frequency-based method, BPE. We apply the morpheme-based tokenization method to both 10-million and 100-million word datasets from the BabyLM Challenge. Our results show that using a morphological tokenizer improves EWoK (basic world knowledge) performance by around 20% and entity tracking by around 40%, highlighting the impact of morphological information in developing smaller language models. We also apply curriculum learning, in which morphological information is gradually introduced during training, mirroring the vocabulary-building stage in infants that precedes morphological processing. The results are consistent with previous research: curriculum learning yields slight improvements for some tasks, but performance degradation in others.

1 Introduction

Large language models (LLMs) have substantially transformed the Natural Language Processing (NLP) domain (Brown et al., 2020). These models leverage vast datasets during pre-training to achieve state-of-the-art performance (Chang et al., 2024). For instance, earlier models, such as GPT-2, were trained on approximately 200 billion tokens (Radford et al., 2019), whereas more recent models, like Llama 3.1, have increased this requirement to over 15 trillion tokens (Grattafiori et al., 2024)[1]. This exponential increase in pre-training data demands highlights the resource-intensive nature of LLMs. Consequently, pre-training such models in low-resource environments poses significant challenges.

In stark contrast, human teenagers master language with exposure to just 100 million words over their whole lifetime (Warstadt et al., 2020a), highlighting a remarkable efficiency gap between human language learning and training LLMs. Therefore, emulating human language acquisition in LLMs could drastically reduce data requirements, making LLMs more viable and effective in resource-constrained settings (Warstadt et al., 2023).

The BabyLM Challenge[2], organized over the past two years (Warstadt et al., 2023; Hu et al., 2024), aims to develop more human-like, data-efficient approaches. To this end, it provides curated child-directed datasets that approximate both the quantity and quality of linguistic exposure experienced by children. These datasets form the basis of a controlled training environment designed to mimic the conditions of early language learning (Capone et al., 2024b). By focusing on such constrained input, the BabyLM Challenge promotes research into models that more closely reflect human-like learning trajectories under limited data regimes (Warstadt et al., 2023; Hu et al., 2024).

In this work, we introduce a morpheme-aware approach where the tokenizer simply splits words into morphologically meaningful units, unlike the other tokenizer methods such as BPE, WordPiece, or SentencePiece (Devlin et al., 2019; Kudo and Richardson, 2018a). This is inspired by child language acquisition, where the vocabulary building stage is followed by morphological and syntactic learning, where relationships between different word forms and words are learned later in the language acquisition (Tomasello, 2003; Clark, 2016).

In addition, we further investigate curriculum learning, in which morphological units are gradually introduced during training. This idea is in-

[1] OpenAI has not disclosed GPT-4's training data, but estimates suggest it was trained on over 13 trillion tokens.

[2] https://babylm.github.io

Proceedings of the First BabyLM Workshop, pages 291–299

spired by child-directed language, where rephrasing is extensively used by employing different morphological forms of the same word in various phrases, and even by emphasizing the bare forms of nouns (i.e. stems) separately. While this approach is especially important for morphologically rich languages, we nonetheless examine it in the context of English, despite its relatively limited morphological complexity.

Our results show that morphological information significantly impacts language models. In particular, our EWoK and entity tracking scores are substantially higher than those obtained with a BPE tokenizer. These results are somewhat surprising, as EWoK measures basic world knowledge rather than a linguistic task. However, the substantial increase in entity tracking aligns closely with the linguistic nature of the task. Curriculum learning positively affects all tasks when using the GPT-BERT architecture (Charpentier and Samuel, 2024), whereas it degrades performance on BLIMP and BLIMP Supplement under the GPT-2 configuration. This is broadly consistent with prior research on curriculum learning (Capone et al., 2024a; Hong et al., 2023), which reports only modest improvements in language model performance.

2 Related Work

Here, we review related work on both tokenization methods and curriculum learning applied to small language models.

Tokenization methods in Small LMs: Bunzeck et al. (2024) use grapheme-based and character-based tokenization along with two different models: grapheme-llama and phoneme-llama. In the phoneme model, they convert the dataset into their phoneme representations, which drastically reduces the vocabulary size. Although the grapheme-based model outperforms the phoneme-based model, the results show that the model can learn the structure of language using only characters as tokens. Analogously, Goriely et al. (2024) use phoneme representations of the dataset. Although the results are slightly lower in language understanding tasks, such phoneme representations have practical advantages, such as in multilingual language modeling.

To our knowledge, this paper is the first to explore morpheme-based tokenization in small language models.

Curriculum Learning Several previous BabyLM Challenge submissions have explored curriculum learning as a strategy to enhance data efficiency and developmental plausibility in language modelling. Diehl Martinez et al. (2023) introduced a curriculum learning framework inspired by infant cognitive development, organizing data to reflect the incremental complexity faced by human learners. Similarly, DeBenedetto (2023) proposed a simple, computationally efficient method for sequencing training data by byte-level difficulty, demonstrating modest gains over random baselines. Oba et al. (2023) approximated natural language acquisition by reordering sentences according to syntactic and lexical complexity, reflecting stages in child language development. Building on the same idea, Hong et al. (2023) used model-based surprisal estimates to dynamically select training examples, aiming to optimize learning trajectories through adaptive data exposure.

In 2024, several approaches continued this trend with more refined techniques. ConcreteGPT (Capone et al., 2024a) implemented a curriculum based on lexical concreteness, training models to first acquire concrete vocabulary before progressing to more abstract terms, thereby mirroring patterns in early word learning.

To the best of our knowledge, no prior small language model has investigated morpheme-based curriculum learning, drawing inspiration from child language acquisition in which vocabulary development precedes the acquisition of morphology and syntax.

3 Methodology

In this study, we investigate morphologically informed tokenization and its impact on language modeling in data-limited contexts, with a particular focus on the BabyLM setting. We employ morpheme-aware tokenization alongside curriculum learning, exploring how these strategies can improve both the efficiency and linguistic generalization of models trained on small corpora. Our approach centers on two key components: (1) the tokenization method and (2) the training regime, with an emphasis on mimicking the stages of early human language development.

3.1 Tokenization Strategies

We compare three tokenization approaches with varying degrees of linguistic awareness: (1) a

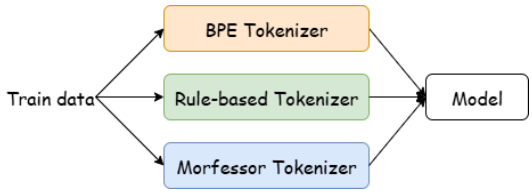


Figure 1: Tokenizers

BPE tokenizer; (2) a rule-based morphological tokenizer; and (3) an unsupervised morphological tokenizer (Morfessor). An overview of the language model, along with the selected tokenizers, is provided in Figure 1.

Byte-pair Encoding (BPE) BPE is a widely used subword tokenization method that segments words based on the frequency of symbol pairs (Gage, 1994; Sennrich et al., 2016). While effective for vocabulary compression and handling out-of-vocabulary words, BPE is agnostic to morphological structure. We include BPE as a standard baseline to evaluate whether morphology-aware tokenizers provide superior advantages in the low-resource BabyLM setting.

Rule-based Tokenizer (Simple) To explicitly incorporate morphological information, we develop a simple rule-based tokenizer that segments words using a predefined list of common English prefixes and suffixes (e.g., 'in', 'un', 'ed', 'ing', 's', etc.). The tokenizer iteratively strips recognized suffixes from the ends of words and prefixes from the beginnings. For example, the word *undoing* is segmented into *un* + *do* + *ing* by identifying 'un' as a prefix, and 'ing' as a suffix using the pre-defined morpheme list. Words shorter than four characters are excluded to reduce oversegmentation. This tokenizer is inspired by early stages of human vocabulary learning, where affix awareness emerges before complex syntactic structures (Tomasello, 2003; Clark, 2016). The method is deterministic, lightweight, and interpretable, making it especially suitable for low-resource conditions. However, it is language-specific and requires a predefined list of morphemes for each target language.

Unsupervised Tokenizer (Morfessor) As a second method for morpheme-based tokenization, we also use the Morfessor (Virpioja et al., 2013), which is an unsupervised morphological analyzer. Unlike BPE, Morfessor produces linguistically plausible segmentations, offering a data-driven but morphology-aware alternative that aligns with our hypothesis about the importance of structured vocabulary building. Moreover, unlike the rule-based morphological tokenizer, it is language-agnostic and can be trained on any language using only a raw corpus.

Table 1 presents sample tokenization outputs for words ranging from morphologically simple to complex, highlighting the differences in segmentation strategies among various tokenizers. As seen, BPE tends to oversegment words depending on their frequency in the dataset, whereas Morfessor and the Simple tokenizer tend to produce longer tokens that better align with the morphemes of the language. However, they remain prone to errors, though they are still better aligned with the morphological structure of words.

3.2 Data

We use the official datasets provided by the BabyLM Challenge: the 10M (Strict-Small) and 100M (Strict) word text-only datasets. These are drawn from a variety of sources, including BNC (Burnard, 2007), CHILDES (Pye, 1994), children's books from Project Gutenberg (Gerlach and Font-Clos, 2020), Simple English Wikipedia, Switchboard (Stolcke et al., 2000), and OpenSubtitles (Lison and Tiedemann, 2016).

We clean the datasets using the cleaning script[3] provided by Timiryasov and Tastet (2023) before training the models.

3.3 Training

We adopt two architectures in our experiments: GPT-2 (Charpentier et al., 2025) and GPT-BERT (Charpentier and Samuel, 2024). Under two architectures, we follow two training approaches in our experiments.

The first training approach is merely built on one of the tokenization methods described above (i.e. BPE, rule-based morphological tokenizer, unsupervised tokenizer), and it involves only one training phase. In the second training approach, we use curriculum learning where the morphological structure of language is gradually introduced during training. In curriculum learning, the first phase corresponds to the vocabulary-building stage in babies, whereas the second phase corresponds to building morphology and syntax, building on top of the vocabulary learned in the first phase.

[3] https://huggingface.co/timinar/baby-llama-58m/blob/main/mrclean.py

Word	BPE	Simple	Morfessor
run	r, un	run	run
dog	d, og	dog	dog
redo	red, o	re, do	re, do
cats	c, ats	cats	cats
jumping	j, ump, ing	jump, ing	jump, ing
played	play, ed	play, ed	played
unhappy	un, happy	un, happy	un, happy
happiness	ha, pp, iness	happi, ness	happiness
friendliness	friend, l, iness	friendli, ness	friendliness
undeniable	un, deniable	un, deniable	undeniable
counterattack	counter, att, ack	counterattack	counter, attack
unbelievably	un, bel, ie, v, ably	un, believab, ly	unbeliev, ably
reconsideration	re, c, ons, ider, ation	re, considera, tion	re, consideration
misunderstanding	m, is, under, standing	misunderstand, ing	misunderstand, ing

Table 1: Comparison of tokenization outputs for selected words by BPE, Simple, and Morfessor tokenizers.

3.4 Evaluation

We evaluate our models through the BabyLM evaluation pipeline (Charpentier et al., 2025). This pipeline consists of six tasks that collectively probe different dimensions of linguistic and cognitive ability.

BLiMP (Warstadt et al., 2020b) measures grammatical knowledge through minimal pair judgments. It consists of minimal pairs of sentences where one is grammatically well-formed and the other is not. EWoK (Ivanova et al., 2024) evaluates basic world knowledge. (Super)GLUE (Wang et al., 2018, 2019) tests general natural language understanding across multiple benchmarks. Entity Tracking (Kim and Schuster, 2023) assesses a model's ability to maintain reference to entities across discourse. Reading (de Varda et al., 2024) evaluates cloze-style reading comprehension. Finally, WUG (Hofmann et al., 2025b) examines the ability of a model to generalize to novel word forms, reflecting morphological productivity. Together, these tasks provide a comprehensive evaluation of models in terms of syntax, semantics, discourse, and generalization, aligning with the developmental plausibility focus of the BabyLM Challenge.

The hidden tasks cover diverse aspects of linguistic competence. WUG_PAST (Weissweiler et al., 2023) tests morphological generalization by correlating model-predicted past tense forms of nonce words with human responses, while WUG_ADJ (Hofmann et al., 2025a) applies the same correlation-based evaluation to adjective nominalization (-ity vs. -ness). COMPS (Misra et al., 2023) probes property inheritance using minimal pairs with nonce concepts, rewarding higher probability for correct sentences. The AoA Benchmark (Chang and Bergen, 2022) tracks surprisal across training to fit learning curves and correlates model-derived acquisition ages with human norms from the MacArthur–Bates CDI[4].

Evaluation metrics We report only zero-shot experiment results on BLiMP, BLiMP Supplement, EWoK, Entity Tracking, and WUG. For reading tasks, we evaluate performance using the coefficient of determination (R^2): Eye Tracking is assessed without spillover, while Self-paced Reading is evaluated with a one-word spillover.

4 Experiments & Results

We use two language model architectures for training the models: GPT-2 (Radford et al., 2019)[5] and GPT-BERT (Charpentier and Samuel, 2024)[6], the winner of the BabyLM 2024. We compare the results with the official results of baselines in BabyLM 2024. The baselines are also based on GPT-2 and GPT-BERT, all using BPE as the tokenizer. GPT-BERT includes two variants, trained with causal language modeling (CLM) and masked next token prediction (MNTP), respectively.

Tokenizer For all tokenizers, we train them on the training corpus with a vocabulary size of $2^{13} = 8192$ in all configurations.

GPT-2 Configuration We adopt the GPT-2 small architecture (Radford et al., 2019), consisting of 12 transformer decoder layers with 12 attention heads,

[4] https://wordbank.stanford.edu/

[5] https://github.com/momergul/babylm-gpt2-baseline

[6] https://github.com/ltgoslo/gpt-bert/

STRICT-SMALL track (10M words)

Model	Tokenizer	BLiMP	BLiMP Supplement	EWoK	Eye tracking	Self-paced Reading	Entity Tracking	WUG
GPT-2	BPE	65.77	62.40	49.82	0.73	0.03	21.93	52.00
GPT-2	SimpleTokenizer	53.04	44.40	53.55	0.74	0.08	40.66	100.00
GPT-2	Morfessor	65.10	49.20	68.45	0.08	0.12	59.65	100.00
GPT-2 (curriculum)	Morfessor	63.19	48.80	69.64	0.09	0.26	59.82	100.00
GPT-BERT	BPE	68.70	61.50	50.40	6.20	**4.45**	25.30	44.50
GPT-BERT	SimpleTokenizer	56.45	49.18	53.18	0.91	0.05	42.18	100.00
GPT-BERT	Morfessor	69.10	50.08	70.01	0.09	0.06	62.17	100.00
GPT-BERT (curriculum)	Morfessor	**72.10**	52.12	**71.15**	0.12	0.36	**63.25**	100
babylm-baseline-10m-gpt2	BPE	66.36	57.07	49.90	8.66	4.34	13.9	52.5
babylm-baseline-10m-gpt-bert-causal	BPE	65.22	59.49	49.47	**9.52**	3.44	30.60	68.00
babylm-baseline-10m-gpt-bert-mntp	BPE	70.36	**63.71**	49.95	9.40	3.37	40.02	57.5

STRICT track (100M words)

Model	Tokenizer	BLiMP	BLiMP Supplement	EWoK	Eye tracking	Self-paced Reading	Entity Tracking	WUG
GPT-2	BPE	75.24	62.80	51.00	2.70	0.43	25.48	47.00
GPT-2	SimpleTokenizer	71.10	48.56	59.17	0.76	0.32	63.10	100.00
GPT-2	Morfessor	64.60	55.20	67.45	0.81	0.28	67.45	100.00
GPT-2 (curriculum)	Morfessor	63.12	49.60	67.82	0.69	0.32	49.47	100.00
GPT-BERT	BPE	79.60	42.60	52.00	6.20	3.05	25.30	45.00
GPT-BERT	SimpleTokenizer	69.18	58.17	69.18	1.05	0.35	67.56	100.00
GPT-BERT	Morfessor	70.12	56.18	69.56	0.98	0.32	**68.48**	100.00
GPT-BERT (curriculum))	Morfessor	73.36	58.43	**71.15**	1.09	0.46	60.21	100.00
babylm-baseline-100m-gpt2	BPE	74.88	63.32	51.67	7.89	3.18	31.51	35.5
babylm-baseline-10m-gpt-bert-causal	BPE	74.56	63.63	51.57	8.80	3.30	30.82	59.00
babylm-baseline-10m-gpt-bert-mntp	BPE	**80.75**	**75.34**	51.77	**9.34**	**3.34**	41.15	55.00

Table 2: Performance of different models across multiple evaluation benchmarks.

a hidden size of 768. The model uses standard initialization (`initializer_range`=0.02) and layer normalization ($\epsilon = 1e^{-5}$). We train for 200k steps with a batch size of 16, using Adam with a learning rate of 5e-5 and 2k warm-up steps. Weight decay is set to zero. The same configuration is used for both strict (100M) and strict-small (10M) data. This configuration contains approximately 124M parameters.

GPT-BERT Configuration We adopt the GPT-BERT architecture (Charpentier and Samuel, 2024) which was the winner of BabyLM 2024. Our implementation follows the configuration reported in the original study, except for the vocabulary size, consisting of 12 transformer layers with a hidden size of 768, weight decay of 0.1, and hidden and attention dropout of 0.1. For the *strict* data (100M), we use 12 attention heads, resulting in approximately 119M parameters, while for the *strict-small* data (10M), we use 6 attention heads with a hidden size of 384, yielding about 30M parameters.

To further limit the computational cost of training, we restrict the context length of the model to 512 tokens in all experiments. All experiments have been carried out locally on one Nvidia H100 GPU.

4.1 Zero-shot Experiments

Table 2 reports results for GPT-2 and GPT-BERT with BPE, SimpleTokenizer, and Morfessor, under both single-stage training and curriculum learning, for the Strict-Small track (10M words) and the Strict track (100M words). Morpheme-based tokenization shows a clear impact on zero-shot tasks, particularly in cognitively demanding settings such as Entity Tracking and EWoK. Models using Morfessor consistently outperform those with BPE or SimpleTokenizer on these benchmarks, often by a substantial margin (e.g., over 20% in EWoK and nearly 40% in Entity Tracking). This improvement likely stems from Morfessor's linguistically informed segmentation, which aligns subword units with meaningful morphological boundaries. By preserving semantic units within words, Morfessor enables the model to better capture entity consistency and relationships, enhancing its ability to track entities across discourse and reason about their attributes. These findings highlight the advantages of morphology-aware tokenization in low-resource settings where semantic richness and structural sensitivity are essential. Interestingly, while BLiMP scores are comparable between BPE and Morfessor, morpheme-based tokenizers perform substantially worse on BLiMP Supplement.

Curriculum learning yields slight improvements across all scores in the GPT-BERT configuration, but results in minor performance degradation with GPT-2. This suggests that the training strategy does not have a uniform effect on performance, but rather interacts differently with specific architectures. The modest gains observed with curriculum learning are consistent with prior research, which has generally reported small improvements from multi-stage training using data blocks of varying difficulty (Capone et al., 2024a; Hong et al., 2023).

5 Conclusion

We showed the effectiveness of using a morpheme-based tokenizer in low resource settings to train a baby language model. Our results show that a morpheme-based tokenizer outperforms BPE for some tasks, such as EWoK and entity tracking by a substantial margin.

We only used GPT-2 and GPT-BERT for the backbone architecture. The results also show that the impact of a tokenizer can be quite different in different architectures. For example, we also investigated curriculum learning using the morphological complexity as the main criterion in a phased training, and the results are different in GPT-2 and GPT-BERT. The morpheme-based tokenizer improves all the scores, including BLIMP, BLIMP Supplement, EWoK, eye-tracking, and entity tracking, when used with the GPT-BERT architecture, whereas curriculum learning does not help as desired when used with the GPT-2 architecture.

Limitations

We showed the effectiveness of a morpheme-based tokenizer for English, a morphologically-poor language. This choice may have hindered the tokenizer's performance, and its application to a morphologically rich language, such as Turkish, could yield significantly different results. In the future, we aim to apply this method to morphologically rich languages in limited-resource settings.

Although we showed the superiority of a morpheme-based tokenizer over a count-based one like BPE, we did not compare it against other methods such as SentencePiece (Kudo and Richardson, 2018b), or character- and word-level tokenizers. Therefore, its relative performance remains to be determined.

Furthermore, our investigation of curriculum learning was limited to morphological complexity. We did not explore syntactic complexity, which, in child language acquisition, is integral to vocabulary building and follows morphological processing.

Ethics Statement

This study was conducted in accordance with ethical guidelines and regulations. We utilized natural speech data extracted from CHILDES (MacWhinney, 2000). This is an open-source corpus that archives natural speech between caregivers and their children. The data are archived without confidential information about the participants, as children are usually given pseudonyms. Following the ACL Policy on Publication Ethics, we used ChatGPT to assist in refining the wording.

Acknowledgements

We wish to acknowledge Tharindu Ranasinghe for stimulating discussions related to this research.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2024. Graphemes vs. phonemes: battling it out in character-based language models. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 54–64, Miami, FL, USA. Association for Computational Linguistics.

Lou Burnard. 2007. Reference guide for the British national corpus (XML Edition).

Luca Capone, Alessandro Bondielli, and Alessandro Lenci. 2024a. ConcreteGPT: A baby GPT-2 based on lexical concreteness and curriculum learning. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 189–196, Miami, FL, USA. Association for Computational Linguistics.

Luca Capone, Alice Suozzi, Gianluca Lebani, and Alessandro Lenci. 2024b. BaBIEs: A benchmark for the linguistic evaluation of Italian baby language models. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 157–170, Pisa, Italy. CEUR Workshop Proceedings.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word Acquisition in Neural Language Models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM Turns 3: Call for papers for the 2025 BabyLM workshop.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Eve V. Clark. 2016. *First Language Acquisition*, 3 edition. Cambridge University Press.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Justin DeBenedetto. 2023. Byte-ranked curriculum learning for BabyLM strict-small shared task 2023. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 198–206, Singapore. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard Diehl Martinez, Zébulon Goriely, Hope McGovern, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 112–127, Singapore. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Martin Gerlach and Francesc Font-Clos. 2020. A Standardized Project Gutenberg Corpus for Statistical Analysis of Natural Language and Quantitative Linguistics. *Entropy*, 22(1).

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Valentin Hofmann, Leonie Weissweiler, David R Mortensen, Hinrich Schütze, and Janet B Pierrehumbert. 2025a. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025b. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Xudong Hong, Sharid Loáiciga, and Asad Sayeed. 2023. A surprisal oracle for active curriculum language modeling. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 259–268, Singapore. Association for Computational Linguistics.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the Second BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, et al. 2024. Elements of World Knowledge (EWOK): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Najoung Kim and Sebastian Schuster. 2023. Entity Tracking in Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018a. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018b. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Brian MacWhinney. 2000. *The CHILDES Project: Tools for analyzing talk. Third Edition*. Lawrence Erlbaum Associates.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual Minimal Pair Sentences for testing Robust Property Knowledge and its Inheritance in Pre-trained Language Models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Miyu Oba, Akari Haga, Akiyo Fukatsu, and Yohei Oseki. 2023. BabyLM challenge: Curriculum learning based on sentence complexity approximating language acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 290–297, Singapore. Association for Computational Linguistics.

Clifton Pye. 1994. The CHILDES project: Tools for analyzing talk.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*, 26(3):339–373.

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby Llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020b. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the Bugs in ChatGPT's Wugs: A Multilingual Investigation into the Morphological Capabilities of a Large Language Model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Do Syntactic Categories Help in Developmentally Motivated Curriculum Learning for Language Models?

Arzu Burcu Güven **Anna Rogers** **Rob van der Goot**
IT University of Copenhagen, Denmark
{argy, arog, robv}@itu.dk

Abstract

We examine the syntactic properties of BabyLM corpus, and age-groups within CHILDES. While we find that CHILDES does not exhibit strong syntactic differentiation by age, we show that the syntactic knowledge about the training data can be helpful in interpreting model performance on linguistic tasks. For curriculum learning, we explore developmental and several alternative cognitively inspired curriculum approaches. We find that some curricula help with reading tasks, but the main performance improvement come from using the subset of syntactically categorizable data, rather than the full noisy corpus. [1]

1 Introduction

Curriculum Learning (CL), a training regimen where the input is ordered from easier to more difficult, has been shown to improve performance of the machine learning algorithms in various scenarios (Soviany et al., 2022). In NLP, the BabyLM challenge (Warstadt et al., 2023), inspired by human efficiency in acquiring language from a small amount of data, has sparked interest in applying CL to small-scale training setups. Most studies in this research area base their curricula on language or syntactic complexity. However, to quantify these complexities they rely on coarse proxies, such as ordering different corpora (Martinez et al., 2023), mean length of utterance (MLU) (Oba et al., 2023) or the average number of syntactic dependents (Mi, 2023). Despite being a popular approach, CL has not consistently led to performance gains in these settings (Hu et al., 2024).

One of the core corpora in CL studies in NLP is CHILDES (MacWhinney, 2000), which consists mostly of interactions between children and adults. It is currently the primary resource for Child Directed Speech (CDS), which is known to exhibit distinct topical, lexical and morphosyntactic features (Gallaway, 1999; Huttenlocher et al., 2002; Soderstrom, 2007). Several studies use CHILDES as a stand-in for developmentally grounded training (Feng et al., 2024; Huebner, 2018; Huebner et al., 2021; Martinez et al., 2023). Surprisingly, although there are many CL studies relying on CHILDES (based on CDS (Huebner, 2018; Huebner et al., 2021), syntactic complexity (Oba et al., 2023; Mi, 2023), or language complexity (Martinez et al., 2023)), its syntactic properties have not been explored in a fine-grained manner in this line of work.

To address the gaps in the literature, namely the lack of concrete curriculum quantification and the limited analysis of CHILDES both in itself and in comparison to other corpora as training data, we propose a syntax-based approach. Our contributions are as follows:

1. We introduce a toolkit[1] to analyze, label, and order training data based on the syntactic properties of each sentence, based on approximately 300 expert-designed tregexes capturing 71% of sentences in CHILDES.

2. We contribute a detailed analysis of the BabyLM corpora for syntactic properties, and we present the analysis of developmentally motivated marco-categories across each subcorpus.

3. For CHILDES, we examine distributions by age group. We find no clear differences that align with the developmental syntactic stages proposed in language acquisition research, and we propose hypotheses for why this might be the case.

4. We train language models on syntactically and developmentally motivated curricula and compare them against baselines. We find that the primary performance gain stems not from CL

[1] https://github.com/arzuburcuguven/syntactic-categorization

Proceedings of the First BabyLM Workshop, pages 300–312

Macro-category	Syntactic Category	Examples
Simple	Subject-Verb	*She runs. She opens the bottle.*
	Adverbs & Possessives	*Try again. She runs fast. She opens your bottle.*
	Prepositions	*Good for you. She runs with her friend.*
	Particle verbs	*Cut it off. She opens up to you.*
	Auxiliaries	*She can run fast. She should open up to you.*
	Negation	*Don't run fast. She should not open up to you.*
	Tense	*You are running fast. She has been opening up to you.*
Complex	Embedded clauses	*Let's go. I know what I need.*
	To-infinitives	*I want to run. I'm going to call you.*
	Linked clauses	*I want to run and smell flowers. I run because I like it.*
	Relative clauses	*The tooth fairy who loves good children*
	Fragments	*Uh, ah yes, umm, not into that*
Interrogatives	Interrogatives	*What? Is that a hat? Does she know what the moon is?*

Table 1: Developmental macro-categories, associated syntactic categories, and example utterances.

Corpus	Genre	Tokens
CHILDES	Child-directed speech	25.9M
BNC Spoken	Spoken English	9.2M
OpenSubtitles	Movie subtitles	25.8M
Switchboard	Telephone conversations	1.6M
Simple Wikipedia	Encyclopedia	17.3M
Gutenberg	Children stories	31.0M

Table 2: Overview of corpora used in this study, with genre and token count after clean-up.

itself, but from using syntactically categorizable data.

5. We utilize our syntactic classification framework to compile syntactically isolated subcorpora, and conduct a study on cross-construction generalization. We observe mixed results: simpler categories do not cross-generalize, whereas more complex categories can improve performance on other complex ones.

2 Methods

Our overall curriculum design is built upon classifying data by syntactic categories, and ordering the classified data according to curricula. We begin by describing the datasets used in this study, followed by the syntactic categories, the categorizing process, and the curriculum design.

2.1 Datasets

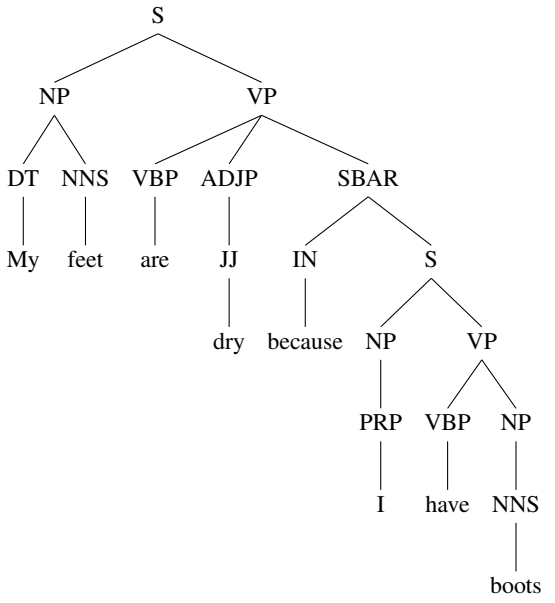
Both the training and the data analysis are conducted on the strict BabyLM dataset (Charpentier et al., 2025). The dataset comprises corpora with diverse properties, including CHILDES as CDS; Switchboard (Godfrey et al., 1992), the spoken portion of the British National Corpus (BNC) (Consortium, 2007), and OpenSubtitles (Lison and Tiedemann, 2016) as adult-directed speech (ADS); and Simple English Wikipedia and Project Gutenberg (children stories) (Gerlach and Font-Clos, 2018) as written text.

We remove speaker labels from all corpora, as the labels decrease the parser accuracy. For CHILDES, we additionally remove annotations and normalize nonstandard expressions. Sentence segmentation is applied to all corpora, and each resulting line is treated as a unit for parsing and extraction. We remove utterances shorter than two tokens. Table 2 summarizes the features and size of each corpus.

2.2 Syntactic Categorization

In order to design the syntactic categories, we examined various resources that classify syntactic phenomena into overarching groups, including typological databases such as Grambank (Lesage et al., 2022), language universals (Croft, 2002), and grammatical frameworks such as dependency relations (De Marneffe et al., 2021), and LinGO Grammar Matrix (Bender et al., 2010). Despite differences in terminology, underlying assumptions, and goals across the frameworks, we curated a set of categories that are at least represented twice among them. We found that the most comprehensive list was presented by Grambank, to which our 13 categories are most closely aligned. We restricted our final set to categories applicable to English. The resulting 13 categories are listed in Table 1. For a further discussion of these categories, see Appendix A.

For parsing the corpora we used Kitaev and Klein (2018)'s a constituency parser for its ease of use and high performance. Data was analyzed using Tregex (Levy and Andrew, 2006), for which



(a) Constituency parse of the sentence "*My feet are dry because I have boots.*"

```
% Subject-verb or intransitive sentence:
(S
  [ <1 (NP <: /NN|DT|PRP|CD|FW|VBG|EX|WP
       /)
  | <1 (NP <1 /NN|DT|PRP|CD|UH|FW|VBG|WP/
         <2 /^NN|DT|PRP|CD|FW|WP/ !<3 __)
  | <1 (NP <1 /NN|DT|PRP|CD|UH|FW|WP/
         <2 /^NN|DT|PRP|CD|FW|VBG|WP/
         <3 /^NN|DT|PRP|CD|FW|WP/ !<4 __)
  ]
  <2 (VP <: /^VB/)
  <3 /^(\.|\.\.\.|!|\?)$/
)
!> __

% Wh-question (e.g., Who is talking to
    you?):
SBARQ<(/WH/$++(/SQ|S/<1(/VB|MD/)<2VP))

% Subordinating conjunction
(e.g., My feet are dry because I have
    boots):
(NP!<<CC)
$++(VP<(/VB/
$++(SBAR<(/IN|WH/$++(S<NP<VP!<<CC)))))
```

(b) Tregex Patterns needed to match the sentence "*My feet are dry because I have boots.*"

Figure 1: Example of syntactic annotation (a) and tregexes (b) used to filter CHILDES

we designed approximately 300 regular expressions targeting the sentences that can be categorized into the 13 syntactic categories. These expressions were crafted by an experienced syntactician with a graduate degree in computational linguistics and six years of professional experience in linguistics. Matches returned by the expressions are saved and reordered to curate corpus subsets. This setup also allows for corpus-specific or cross-corpus categorization. Extracted data can also be used to create filtered training data, for example, by excluding fragments or only including relative clauses.

Figure 1a shows a constituency tree of a complex sentence and Figure 1b shows examples of Tregex patterns used to match the syntactic trees to different categories.

To the best of our knowledge, our Tregex patterns constitute the most extensive syntactic analysis of CHILDES to date; prior parsing studies used much smaller subsets (65k-236k tokens; (Sagae et al., 2007; Liu and Prud'hommeaux, 2023; Yang et al., 2025)). Even so, it categorizes only 71% of sentences in the English portion of CHILDES, primarily because of the long tail of rare that would be impossible to fully cover with Tregexes and presence of noisy disfluencies (stutters, restarts, fillers), e.g., "y you know b build this like real big thing to hold t planets from colliding together."

2.3 Curriculum

Most studies on language acquisition in English-speaking children focus on a specific syntactic phenomenon or developmental period. For instance, the seminal work by Brown (2013) describes the acquisition of a variety of phenomena such as tense, possessives, and auxiliaries, yet omits others such as interrogatives and conjunctions. Similarly, Braine and Bowerman (1976) focus exclusively on the first word combinations. Many studies approach acquisition from a universalist perspective, highlighting similarities among different language speakers (Slobin, 1987).[2]

However, to create a syntactically grounded developmental curriculum, we need a more comprehensive framework representing a wider range of phenomena. To this end, we adopted the developmental stages proposed by Friedmann and Reznick (2021), based on observations of 54 Hebrew-speaking children aged 1.5 to 6 years. These stages have also been applied to English to examine whether similar learning trajectories are also

[2]For numerous language-specific studies, see the series *The Crosslinguistic Study of Language Acquisition* (ed. D. I. Slobin).

observed in the learning behavior of LMs (Evanson et al., 2023).

Friedmann and Reznick (2021) identify three main stages in syntactic development: the first stage corresponds to simple subject–verb constructions, the second to interrogatives, and the third to relative clauses and embedded structures such as infinitives. We adopt these three stages as the basis for our main curriculum, labeling them as simple, interrogative, and complex. The 13 syntactic categories are mapped to these macro-categories as shown in Table 1.

We stress that this is only one possible hypothesis about how an effective curriculum could be constructed, and any conclusions would be made only with respect to it rather than developmentally motivated CL in general.

2.4 Evaluation

We evaluated our models using the shared BabyLM evaluation pipeline (Charpentier et al., 2025). Model evaluation was conducted on the full test set, with the exception of the Age of Acquisition (AoA) Evaluation Benchmark (Chang and Bergen, 2022). The evaluation suite includes BLiMP (Warstadt et al., 2020), EWoK (Ivanova et al., 2024), COMPS (Misra et al., 2023), (Super)GLUE (Wang et al., 2018), Entity Tracking (Kim and Schuster, 2023), WUG_ADJ (Hofmann et al., 2024), WUG_PAST (Weissweiler et al., 2023), and Reading (self-paced and eye-tracking) (de Varda et al., 2024).

BLiMP is a linguistic evaluation suite and BLiMP Supplement includes tasks specifically designed for BabyLM. COMPS and EWoK are world-knowledge datasets: COMPS focuses on immutable properties and their inheritance to subordinate concepts, whereas EWoK targets more dynamic, context-dependent properties. The Entity Tracking task assesses a model's ability to follow the states of discourse entities. WUG_ADJ evaluates adjective nominalization on nonce words, while WUG_PAST assesses past-tense formation on nonce words. The Reading task measures the alignment between LM predictions and human processing through comparison with reading times. Lastly, GLUE is used for fine-tuning evaluation.

2.5 A Closer Look into Datasets

This section provides an exposition of syntactic properties of corpora under study. First, we compare BabyLM sub-corpora and discuss differences in their distributions. Second, we examine age-ordered CHILDES to see whether syntactic distributions follow a developmental trajectory.

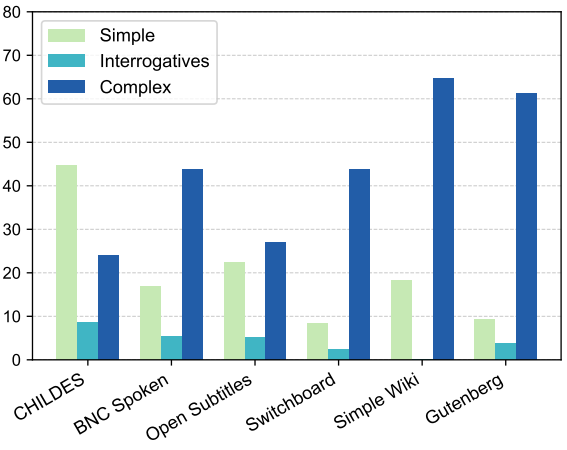


Figure 2: Distribution of macro-categories across corpora. Y-axis shows the percentage of sentences in each macro-category relative to the total number of sentences in the corpus.

2.5.1 Differences Among Corpora

In Figure 2 we present the ratio of sentences that fall under each of the macro-categories for six different corpora. Here we can see the effect of corpus genre clearly, CHILDES, being the only example of CDS differs markedly from other BabyLM corpora: Simple constructions and interrogatives account for 49% of CHILDES, compared to 10.7–27.2% in the other corpora. Among ADS corpora, BNC Spoken and Open Subtitles lean toward simpler language (16.1% simple and 5.2% interrogatives for the former; 22.0% simple and 5.2% interrogatives for the latter), whereas Switchboard has the lowest ratio of simple sentences (8.3%) and a distribution more closely aligned with text corpora.

Among written corpora, Simple English Wikipedia has the lowest proportion of interrogatives (0.04%), while Project Gutenberg is the most complex-leaning corpus, containing the highest proportion of complex sentences (59.8%).

These distributions can be useful in interpretation of model performance as identifying which constructions are rare or overrepresented in the training data provides insight into model performance across different constructions. For instance, Huebner et al. (2021) suggest that the high frequency of questions in CHILDES may explain why models trained on it perform better on interrogatives. Indeed, among the corpora analyzed here, it has the highest proportion of interrogatives (7.8%).

Padovani et al. (2025) compare models trained on CHILDES and Wikipedia. They evaluate the models on various agreement pairs and find that models trained on Wikipedia tend to perform better. This result is aligned with the distributions as relative clauses, which are one of the most challenging agreement distractors, are very scarce in CHILDES, amounting to only 0.8% of the data whereas in Simple English Wikipedia, relative clauses account for the 11.5% of the data, providing much richer training signal in terms of distractors.

2.5.2 Age-Ordered CHILDES

It is well-established that CDS is markedly different from ADS. One reason for this divergence is that adults adjust the syntactic complexity of their speech to match the child's level of comprehension (Snow, 1972; Iii and Marquis, 1977). Prior studies show that the syntactic complexity of CDS tends to increase over time, and that these changes in input correlate with children's language growth (Huttenlocher et al., 2010; Silvey et al., 2021). Given the relationship between CDS and the child's linguistic ability, we hypothesized that the age-ordered CHILDES would reflect the syntactic development of children.

Few studies have examined the differences between the age groups within CHILDES. Among them, the most relevant to our work is Bunzeck and Diessel (2025), which utilizes the morphological annotations within CHILDES with a regex-based parser, and assigns each sentence to one syntactic group among six: subject-verb constructions, interrogatives, imperatives, copular clauses, complex sentences and fragments. Their results show a subtle tendency toward interrogatives in the earlier age groups and subject-verb constructions in the older ones.

We plot the macro-categories over age groups in Figure 3, the full results on the fine-grained categories are reported in Appendix A, Figure 5. Our results do not reveal a clear developmental pattern across age groups. In line with Bunzeck and Diessel (2025)'s results, there is a subtle tendency toward interrogatives in the earlier age groups, highest being 17.5% with 3 to 4 age group. Subject-verb constructions, on the other hand, follow a non-linear trajectory, they peak at 48.9% in between the ages of 1 to 2, then decrease and increase again between the ages of 5 and 6. Excluding the preverbal group, complex constructions start from 14.6% in 1 to 2 ages and increase to 23.8% at 5 to 6. In

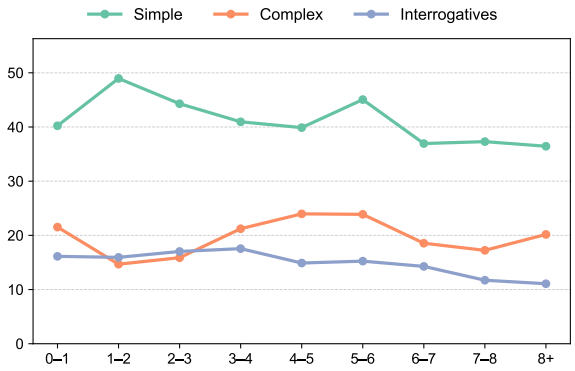


Figure 3: Distribution of macro-categories across age-ordered CHILDES. X-axis: age groups; Y-axis: percentage of sentences per macro-category.

agreement with Soderstrom (2007)'s findings, the preverbal segment of the corpus is syntactically distinct with a surprisingly high proportion of complex constructions (21%).

Our results suggest that CHILDES as a whole may not exhibit strong syntactic differentiation by age. Several factors likely contribute to this counter-intuitive outcome. The age groups aggregate data from 58 subcorpora, each containing transcripts from multiple children. Since children reach developmental milestones at individual rates (Bates et al., 2019), it may be more informative to track syntactic development longitudinally for each child, as in Brown (2013). Socioeconomic status and dialect are also known to affect language complexity (Huttenlocher et al., 2002). Lastly, CHILDES transcripts come from different sessions, such as free play and book reading, which are known to differ in their syntactic characteristics (Bunzeck and Diessel, 2025).

3 Experiments

For both CL and generalization studies, we trained a model with the GPT-2 small architecture (124M parameters) (Radford et al., 2019) from scratch using the Hugging Face Transformers library (Wolf et al., 2020). Hyperparameters are detailed in Appendix B.

3.1 Experiment 1: Curriculum

3.1.1 Methodology

This section describes experiments in which training sets are organized according to different curriculum approaches. The research question we address is "Does training on a developmentally motivated syntactic curriculum improve LM per-

Condition	BLIMP	SUPPLEMENT	EWOK	COMPS	GLUE
B1	70.24 ± 0.17	**57.66 ± 0.10**	**50.53 ± 0.28**	**52.94 ± 0.49**	57.12 ± 0.53
B2	**71.13 ± 0.62**	52.98 ± 0.70	50.27 ± 0.18	51.74 ± 0.39	57.80 ± 0.74
C1	69.88 ± 0.86	54.43 ± 2.04	50.08 ± 0.20	51.54 ± 0.58	**57.83 ± 0.51**
C2	70.45 ± 0.72	55.85 ± 0.63	50.20 ± 0.18	51.19 ± 0.41	57.45 ± 0.41
C3	70.98 ± 0.52	54.28 ± 0.29	50.06 ± 0.22	51.75 ± 0.80	57.62 ± 0.58
C4	70.03 ± 0.60	53.09 ± 0.97	49.94 ± 0.26	51.31 ± 0.09	57.80 ± 0.35
C5	70.44 ± 0.48	54.40 ± 0.89	50.19 ± 0.16	51.36 ± 0.43	57.61 ± 0.70

Table 3: Mean ± SD (over seeds) for BLiMP, Supplement, EWOK, COMPS, and GLUE. Best per column in **bold**.

Condition	ENTITY	WUG_ADJ	WUG_PAST	READING_SPR	READING_ET
B1	20.70 ± 6.09	51.10 ± 7.76	**2.28 ± 7.98**	0.04 ± 0.05	0.42 ± 0.08
B2	**41.24 ± 1.21**	**68.87 ± 1.63**	-15.81 ± 6.08	0.14 ± 0.05	0.48 ± 0.17
C1	32.34 ± 6.65	65.12 ± 1.67	-19.89 ± 10.64	**0.17 ± 0.05**	0.64 ± 0.16
C2	31.68 ± 8.75	62.06 ± 3.70	-22.81 ± 5.26	0.15 ± 0.07	**0.65 ± 0.12**
C3	38.76 ± 2.53	67.51 ± 1.10	-15.71 ± 6.15	0.08 ± 0.04	0.42 ± 0.06
C4	37.76 ± 3.71	66.84 ± 3.28	-24.32 ± 3.86	0.05 ± 0.03	0.35 ± 0.08
C5	37.83 ± 4.43	65.52 ± 4.60	-22.73 ± 2.13	0.12 ± 0.07	0.39 ± 0.04

Table 4: Mean ± SD (over seeds) for entity tracking, WUG, and reading metrics. WUG_PAST column shows correlation results multiplied by 100. Best per column in **bold**.

Cond.	Tokens	Data order
B1	131M	Random
B2	77M	Random
C1	77M	S→I→C
C2	77M	S→C
C3	77M	S→C (gradual)
C4	77M	80% SIC, 20% Mixed
C5	77M	20% Mixed, 80% SIC, 20% Mixed

Table 5: Summary of training conditions. S=Simple, I=Interrogatives, C=Complex.

formance compared to random ordering or other curriculum variants?" To this end we train seven models: two baselines (B1, B2) and five curriculum variants (C1–C5). Table 5 summarizes all training conditions.

The baselines are B1, the full BabyLM corpus in random order, and B2, an extracted subset of BabyLM corpus containing the union of all syntactically categorized data in random order. C1 (developmental curriculum, Section 2.3) groups the syntactically categorized training data into simple, interrogative, and complex stages, shuffling within each stage before concatenating them to form the final corpus.

To contrast with the developmentally grounded approach, we also devise several alternative curricula. In the simple-to-complex curriculum (C2), we categorize each syntactic structure as either simple or complex based on the presence of nested embedding. We then concatenate these two subgroups. In C3, we use the same simple and complex division described above but interleave them such that the dataset starts from only simple examples, progresses to a balanced dataset and ends with only complex examples. To achieve this, we employ a probabilistic sampling function that decreases the probability of sampling from the simple dataset and increase the probability of sampling from the complex dataset over the course of the sampling process.

The last two CL approaches are inspired by the Learn–Focus–Review (LFR) strategy of Prakriya et al. (2025), a cognitively inspired dynamic learning paradigm. In the initial learn phase, models see a portion of randomly sampled training data. In the focus phase, more challenging portions of the data are clustered, and in the review phase, the remaining data is reintroduced to prevent forgetting. For C4, 20% of the syntactically labeled data is held out, the remaining 80% is constructed as in C1, and the held-out portion is appended as a review at the end. For C5, 40% of the data is held out, 60% is constructed as in C1, and the held-out portion is split in half, with one half appended to the beginning and the other half to the end of the corpus.

3.1.2 Results

We report averaged results over four seeds on the BabyLM test suite in Table 4 and Table 3. While

Condition	Hypernym	QA_easy	QA_tricky	SubjAuxInv	Turn_taking
B1	48.99 ± 0.35	**55.47 ± 2.71**	**39.55 ± 1.04**	84.02 ± 1.32	**60.27 ± 2.17**
B2	49.82 ± 0.50	49.61 ± 2.66	27.88 ± 3.39	87.68 ± 1.81	49.91 ± 1.94
C1	50.27 ± 0.68	52.34 ± 4.51	36.21 ± 2.29	83.77 ± 5.53	49.55 ± 0.45
C2	49.94 ± 1.08	52.73 ± 3.46	38.03 ± 3.14	**87.95 ± 0.66**	50.62 ± 1.28
C3	50.23 ± 1.52	53.90 ± 1.57	29.39 ± 1.61	87.20 ± 1.80	50.62 ± 0.68
C4	49.47 ± 1.10	50.00 ± 2.21	30.00 ± 1.60	85.88 ± 0.70	50.09 ± 1.18
C5	**50.62 ± 0.91**	52.73 ± 1.97	31.06 ± 1.94	88.54 ± 1.07	49.02 ± 1.38

Table 6: Mean ± SD over seeds for UID subtasks. Best per column in bold.

curriculum learning offers some task-specific benefits, the main finding is that models trained on parsed and categorized data perform on par with the B1 baseline despite requiring 40% fewer training steps. B1 still leads on BLiMP Supplement, EWOK, COMPS and WUG_PAST, though the EWOK and COMPS margins are small.

The difference in BLiMP Supplement scores may stem from a preprocessing decision: to make our training data parser-compatible, we removed speaker labels. As a result only the B1 model, which was trained on the whole BabyLM corpus, was shown examples with speaker labels. As shown in Table 6 (Appendix B), B1's higher Supplement score is concentrated in three subcategories, *QA_easy*, *QA_tricky*, and *turn-taking*; each containing speaker labels. Since in the main BLiMP benchmark and other Supplement categories the other models outperform B1, this suggests that presence of the speaker labels likely accounts for the observed gap.

The difference in performance on WUG_PAST is more difficult to interpret. A qualitative analysis of the predictions shows that B1 models tend to apply regular inflection to wug words more often, aligning more closely with human data. In contrast, the other models more frequently produce irregular inflections, correlating negatively with the baseline. For the WUG_ADJ task, however, B1 underperforms compared to all other models. One possible explanation is that cleaner data makes models more attentive to irregularities. This may be an advantage in tasks with a constrained prediction space, such as selecting from a limited set of adjective nominalizers, but a disadvantage in open-set tasks like WUG_PAST.

For GLUE, entity tracking, and reading tasks, models trained on categorized data outperform the B1 models. Especially for reading tasks, both self-paced reading and eye-tracking, curriculum models C1 and C2 show the highest performance, suggesting that the curriculum approaches can provide a signal that shortens the gap between human and machine processing.

Category	Constructions
Subject-Verb	Subject-Verb patterns
Modifier	Adverbs, Possessives, and Prepositions
Verbal	Particle verbs, Auxiliaries, Negation, and Tense
Embedded C.	Small clauses, reported speech
Infinitives	Infinitives
Linked Clauses	Coordination, Subordination
Relative Clauses	Relative Clauses
Interrogatives	Yes/no, wh-questions

Table 7: Syntactic category groups used in the generalization study and their corresponding constructions.

3.2 Experiment 2: Generalization

3.2.1 Methodology

In this experiment, each model is trained on a single category and evaluated on eight validation sets corresponding to the distinct eight categories given in Table 7. We approach this task as a generalization study, using the perplexity values as a proxy for models' ability to learn both the category they trained on and the remaining seven unseen categories.

For each group, we sample 2M tokens for training and 200K tokens for validation from the syntactically classified portion of BabyLM corpus. Sampling is restricted to sentences matching the target group's criteria. We train GPT-2 small models from scratch on each subset for one epoch. All results are averaged over five random seeds.

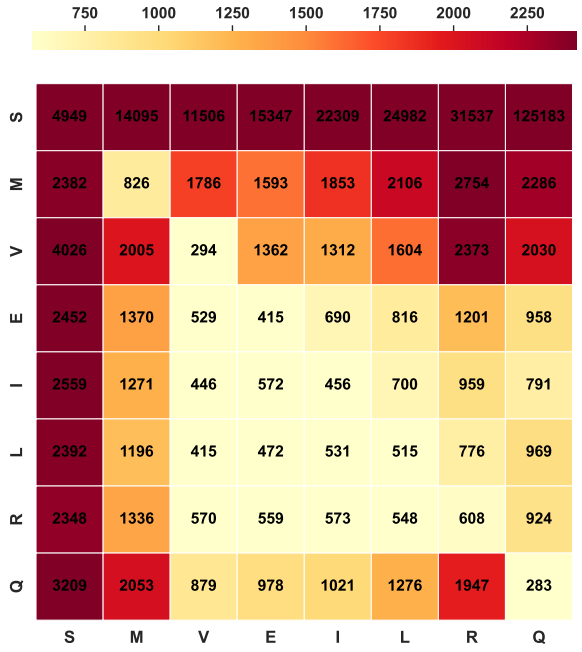


Figure 4: Cross-subset validation perplexity heatmap. Rows = training subset; columns = evaluation subset. Abbreviations: S=SVX, M=Modifiers, V=Verbal, E=Embedded, I=Infinitives, L=Coordination, R=Relative, Q=Question. Cell values are validation perplexities (lower is better).

3.2.2 Results

In Figure 4 we report mean perplexity values across seeds. As expected, each model achieves its lowest perplexity when evaluated on the same syntactic category it was trained on (diagonal entries). Off-diagonal values indicate cross-category generalization.

Performance patterns vary across categories. The *Subject–Verb* group (SVX) shows the largest drop in both in-category and cross-category performance, likely due to the high frequency of single-word (e.g., "Run!") and fragmentary utterances (e.g., "all gone"). The *Verbal* and *Modifier* groups also generalize poorly. Models trained on *questions*, despite the data exhibiting unique syntactic patterns such as subject auxiliary inversion, generalize better than those trained on *Subject–Verb*, *Verbal* and *Modifier* constructions. Models trained on complex constructions tend to generalize better to other complex categories. The *Coordination*-trained model exhibits the strongest overall generalization, with the lowest mean off-diagonal perplexity (962.20) and the lowest perplexity on the mixed test set (574.2).

Overall perplexities remain high, and there is limited evidence for genuine syntactic generalization, particularly from simpler to more complex categories. Prior work demonstrating such transfer with transformer architectures typically relies on synthetic datasets with tightly controlled syntax and vocabulary (Murty et al., 2023; Ahuja et al., 2025; Someya et al., 2024). Our subsets are selected by syntactic criteria but retain naturalistic variation in sentence form and vocabulary. These results highlight the difficulty of isolating syntactic generalization in naturalistic data and suggest that stricter control of lexical and structural properties may be necessary for clearer conclusions.

4 Conclusion

This study contributes the most detailed syntactic analysis of BabyLM data to date, implemented as an open-source toolkit for analysing, labeling and ordering training data.[1] This enabled both modeling experiments and a systematic analysis of syntactic patterns in CHILDES, where, counter-intuitively, we find no clear differences in distributions that would align with syntactic stages proposed in language acquisition research. Likewise, we find that developmentally motivated curriculum has a modest effect in language model training, compared to simply training the models on a subset of training data filtered to only syntactically categorizable sentences.

Efficient curriculum learning for language models that is inspired by human learning stages remains an elusive goal. The results of this study suggest that continued focus solely on syntax may be counter-productive, and that the noise in popular resources such as CHILDES may by itself have an outsized effect in studies relying on it.

Limitations

We note the following limitations of this study:

1. We did not observe developmental patterns in the aggregated CHILDES data, but our analysis did not extend to a more fine-grained level where confounding factors could be mitigated.

2. Our syntactic categorization covered 71% of the BabyLM; some of the remaining gap is attributable to our data cleaning practices, but a portion remains unexplained.

3. The absence of clear effects from CL or generalization may stem from several factors, and this study does not establish which ones are

the most relevant. It is possible that isolating syntactic properties alone could be insufficient, or our method of isolation may not capture the most relevant distinctions. Alternatively, the targeted developmental progression and generalization may not be reproducible with the transformer architecture or training conditions used.

References

Kabir Ahuja, Vidhisha Balachandran, Madhur Panwar, Tianxing He, Noah A. Smith, Navin Goyal, and Yulia Tsvetkov. 2025. Learning Syntax Without Planting Trees: Understanding Hierarchical Generalization in Transformers. *Transactions of the Association for Computational Linguistics*, 13:121–141. Place: Cambridge, MA Publisher: MIT Press.

Elizabeth Bates, Philip Dale, and Donna Thal. 2019. Individual Differences and their Implications for Theories of Language Development. pages 95–151.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar Customization. *Research on Language and Computation*, 8(1):23–72.

Martin D. S. Braine and Melissa Bowerman. 1976. Children's First Word Combinations. *Monographs of the Society for Research in Child Development*, 41(1):1.

Roger Brown. 2013. *A First Language: The Early Stages*. Harvard University Press.

Bastian Bunzeck and Holger Diessel. 2025. The richness of the stimulus: Constructional variation and development in child-directed speech. *First Language*, 45(2):152–176. Publisher: SAGE Publications Ltd.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word Acquisition in Neural Language Models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM Turns 3: Call for papers for the 2025 BabyLM workshop. *arXiv preprint*. ArXiv:2502.10645 [cs].

BNC Consortium. 2007. British national corpus, XML edition. Literary and Linguistic Data Service.

William Croft. 2002. *Typology and Universals*, 2 edition. Cambridge University Press.

Marie-Catherine De Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, pages 1–54.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 English sentences, aligned with existing EEG and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Linnea Evanson, Yair Lakretz, and Jean Rémi King. 2023. Language acquisition: do children and language models follow similar learning stages? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12205–12218, Toronto, Canada. Association for Computational Linguistics.

Steven Y. Feng, Noah Goodman, and Michael Frank. 2024. Is Child-Directed Speech Effective Training Data for Language Models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22055–22071, Miami, Florida, USA. Association for Computational Linguistics.

Naama Friedmann and Julia Reznick. 2021. Stages rather than ages in the acquisition of movement structures: Data from sentence repetition and 27696 spontaneous clauses. *Glossa: a journal of general linguistics*, 39(1).

Clare Gallaway, editor. 1999. *Input and interaction in language acquisition*, 1. publ. [transferred to digital reprinting] edition. Cambridge University Press, Cambridge.

Martin Gerlach and Francesc Font-Clos. 2018. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Preprint*, arXiv:1812.08092.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: telephone speech corpus for research and development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 517–520 vol.1, San Francisco, CA, USA. IEEE.

Valentin Hofmann, Leonie Weissweiler, David Mortensen, Hinrich Schütze, and Janet Pierrehumbert. 2024. Derivational Morphology Reveals Analogical Generalization in Large Language Models. *arXiv preprint*. ArXiv:2411.07990 [cs].

Yaling Hsiao, Nicola J. Dawson, Nilanjana Banerji, and Kate Nation. 2023. The nature and frequency of relative clauses in the language children hear and the language children read: A developmental cross-corpus analysis of English complex grammar. *Journal of Child Language*, 50(3):555–580.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the Second BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. *arXiv preprint*. ArXiv:2412.05149 [cs].

Philip Huebner. 2018. Order matters: Distributional properties of speech to young children bootstrapslearning of semantic representations. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 40(0).

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning More Grammar With Small-Scale Child-Directed Language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Janellen Huttenlocher, Marina Vasilyeva, Elina Cymerman, and Susan Levine. 2002. Language input and child syntax. *Cognitive Psychology*, 45(3):337–374.

Janellen Huttenlocher, Heidi Waterfall, Marina Vasilyeva, Jack Vevea, and Larry V. Hedges. 2010. Sources of Variability in Children's Language Growth. *Cognitive psychology*, 61(4):343–365.

John Neil Bohannon Iii and Angela Lynn Marquis. 1977. Children's Control of Adult Speech. *Child Development*, 48(3):1002.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, et al. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Najoung Kim and Sebastian Schuster. 2023. Entity Tracking in Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency Parsing with a Self-Attentive Encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Jakob Lesage, Hannah J. Haynie, Hedvig Skirgård, Tobias Weber, and Alena Witzlack-Makarevich. 2022. Overlooked Data in Typological Databases: What Grambank Teaches Us About Gaps in Grammars. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2884–2890, Marseille, France. European Language Resources Association.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Zoey Liu and Emily Prud'hommeaux. 2023. Data-driven Parsing Evaluation for Child-Parent Interactions. *Transactions of the Association for Computational Linguistics*, 11:1734–1753.

Brian MacWhinney. 2000. The childes project. 1: Transcription format and programs. Erlbaum, Mahwah. Num Pages: 159.

Richard Diehl Martinez, Zebulon Goriely, Hope McGovern, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB: Curriculum Learning for Infant-inspired Model Building. *arXiv preprint*. ArXiv:2311.08886 [cs].

Maggie Mi. 2023. Mmi01 at The BabyLM Challenge: Linguistically Motivated Curriculum Learning for Pretraining in Low-Resource Settings. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 269–278, Singapore. Association for Computational Linguistics.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual Minimal Pair Sentences for testing Robust Property Knowledge and its Inheritance in Pre-trained Language Models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher Manning. 2023. Grokking of Hierarchical Structure in Vanilla Transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 439–448, Toronto, Canada. Association for Computational Linguistics.

Miyu Oba, Akari Haga, Akiyo Fukatsu, and Yohei Oseki. 2023. BabyLM Challenge: Curriculum learning based on sentence complexity approximating language acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 262–269, Singapore. Association for Computational Linguistics.

Francesca Padovani, Jaap Jumelet, Yevgen Matusevych, and Arianna Bisazza. 2025. Child-Directed Language Does Not Consistently Boost Syntax Learning in Language Models. *arXiv preprint*. ArXiv:2505.23689 [cs].

Neha Prakriya, Jui-Nan Yen, Cho-Jui Hsieh, and Jason Cong. 2025. Accelerating large language model pretraining via LFR pedagogy: Learn, focus, and review. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 268–290, Vienna, Austria. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Kenji Sagae, Eric Davis, Alon Lavie, Brian MacWhinney, and Shuly Wintner. 2007. High-accuracy Annotation and Parsing of CHILDES Transcripts. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 25–32, Prague, Czech Republic. Association for Computational Linguistics.

Catriona Silvey, Özlem Ece Demir-Lira, Susan Goldin-Meadow, and Stephen W. Raudenbush. 2021. Effects of Time-Varying Parent Input on Children's Language Outcomes Differ for Vocabulary and Syntax. *Psychological Science*, 32(4):536–548.

Dan I. Slobin, editor. 1987. *The crosslinguistic study of language acquisition, Vol. 1: The data*. Erlbaum, Hillsdale, NJ.

Catherine E. Snow. 1972. Mothers' Speech to Children Learning Language. *Child Development*, 43(2):549.

Melanie Soderstrom. 2007. Beyond babytalk: Re-evaluating the nature and content of speech input to preverbal infants. *Developmental Review*, 27(4):501–532.

Taiga Someya, Ryo Yoshida, and Yohei Oseki. 2024. Targeted syntactic evaluation on the Chomsky hierarchy. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15595–15605, Torino, Italia. ELRA and ICCL.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum Learning: A Survey. *International Journal of Computer Vision*, 130(6):1526–1565.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiulin Yang, Zhuoxuan Ju, Lanni Bu, Zoey Liu, and Nathan Schneider. 2025. UD-English-CHILDES: A Collected Resource of Gold and Silver Universal Dependencies Trees for Child Language Interactions. *arXiv preprint*. ArXiv:2504.20304 [cs].

A Category Details

Below, we list our categories ordered by an increasing number of terminals and combinatorial possibilities. We start from simple noun phrases (NP), verb phrases (VP), adjective phrases (ADJP) and Subject-Verb constructions that can be built with them. For the categories with simpler constructions without any nested structures, the Tregex patterns match entire sequences and tightly constrain the contents of each node to exclude any complex expansions within the tree. For the more complex categories, we switch to partial matching, without constraining the preterminal nodes.

- *Subject-Verb Constructions*: For the sake of readability we use the term Subject-Verb Constructions, but the structures included are intransitive sentences (SV), transitive sentences (SVO), imperatives and copular sentences (SVC). Preterminals included in this category are simple NPs, VPs and ADJPs that have limited amount of nodes and no nested structures under them. Along with the well formed structures, we include sequences that consist

of phrases such as *Beautiful girl, the doll, all toys, love you Baby* etc. For the following categories up to the interrogatives, the sentence structures are limited to the ones described here.

- *Possessives and Adverbials*: For this category, we add POS and ADV preterminals to the former group. The NPs are extended to include possessives e.g., *The girl's hat is beautiful*. Adverbial phrases are allowed both under VPs and directly under the S node.

- *Prepositions*: Phrases headed by PPs (*at the table*), NPs governing over PPs (*the girl with the blue ribbon*), ADJPs governing over PPs (*good for you*) and VPs governing over PPs (*walk to me*) are included both as standalone phrases and as participants in the SVX structures.

- *Particles*: VP categories are extended to include particle verbs (*take off, put on*). This category forms one of the smallest categories in terms of how many sentences it captures, along with auxiliaries and tense.

- *Auxiliaries*: Here we repeat all the canonical sentence types from the former categories, SVX, SVX with adverbs, SVX with PPs and so on and modify the VPs to govern over an auxiliary.

- *Negation*: The scope is again limited to all the canonical sentence types from the former categories and VPs are modified to govern over the negation particle.

- *Tense*: Although we have not differentiated between simple present or simple past tenses in the former categories, the more complex tenses such as progressive and perfective require a specific VP category. Again, we repeat all the canonical sentence types from the former categories, and modify the VPs to allow for the capture of complex tenses.

- *Interrogatives*: Here we include different types of interrogatives: Yes/no questions (*Is she coming?*), Wh-questions (*What is she doing?*), tag questions (*She doesn't know, does she?*) and question fragments (*What?, Did she?*).

Hyperparameter	Value
Model type	GPT-2 small
Parameters	124M
Vocabulary size	50,257
Context size	1024
Dropout	0.1
Learning rate	1.88×10^{-4}
Scheduler	Linear
Weight decay	0
Epochs	1
Batch size	8
Optimizer	AdamW

Table 8: Training hyperparameters for GPT-2 small

- *Embedded Clauses*: This group captures a variety of nested structures in which at least two predicates are present. This includes let-constructions such as *let me go*, causatives (*I will make him bite mommy*) and small clauses (*I think you can fix it*).

- *Infinitives*: This category captures the to-infinitives and gerunds e.g., *She wants to drink from her cup.*

- *Clause Linking*: Here we include coordinating conjunctions (*She ate an apple but the apple was rotten*) and subordinating conjunctions (*My feet are dry because I have boots.*).

- *Relative Clauses*: This category is adapted from Hsiao et al. (2023), which includes relative clauses of subject (*The man who kicked the ball*), object (*the fun I had*) and passive (*the houses that were built*) types.

- *Fragments*: While we allow phrase level constructions when they represent a well formed phrase, malformed phrases and interjections fall into this group.

B Model Details

We tuned hyperparameters with a sweep: learning rate sampled log-uniformly in $[5\times10^{-6},\ 5\times10^{-4}]$ and per-device train batch size $\in \{8, 16, 32\}$; the best model was selected by validation-set perplexity. Remaining hyperparameters were taken from Radford et al. (2019). The full set of hyperparameters is shown in Table 8.

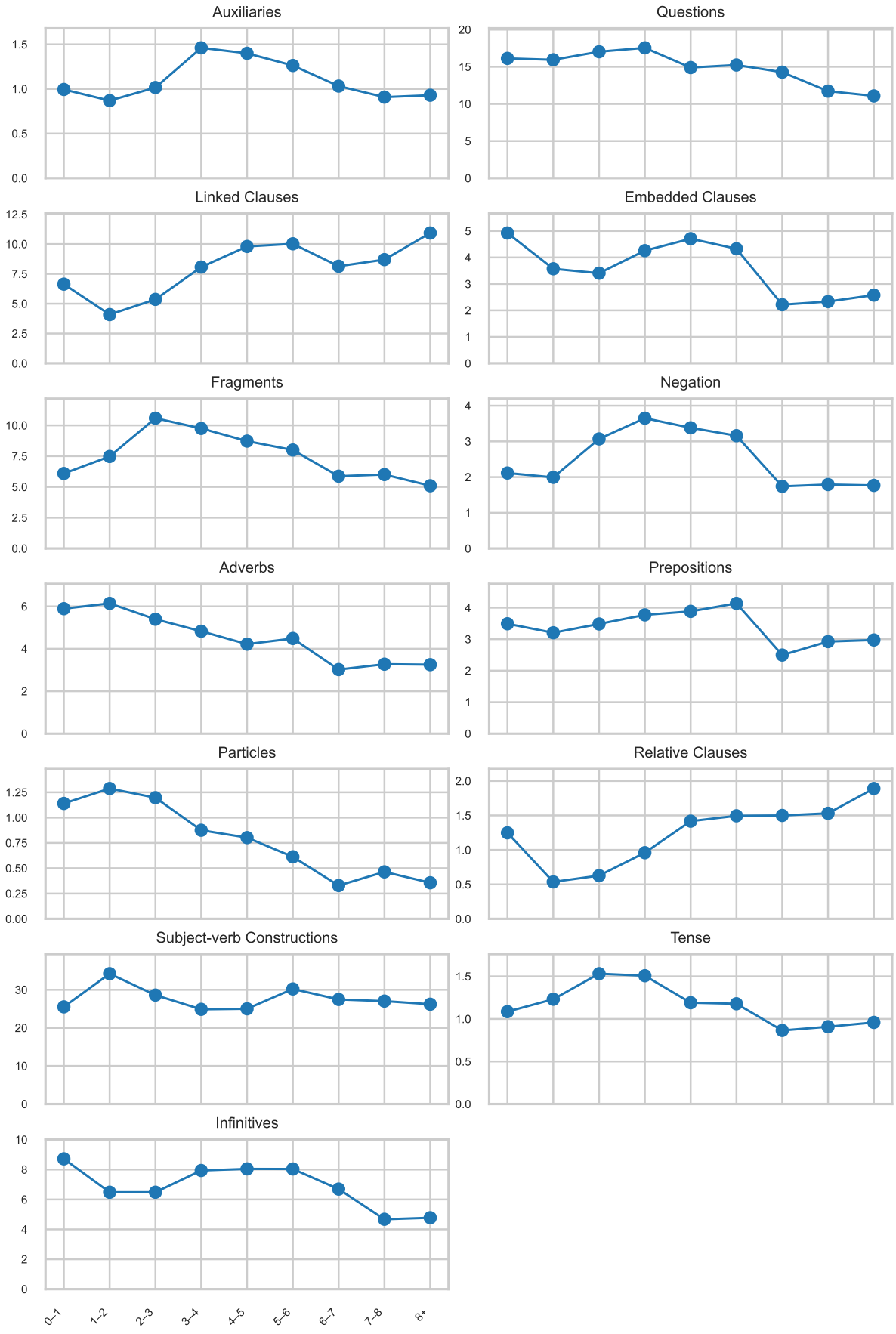


Figure 5: Percentage distribution of syntactic categories across age groups in CHILDES.

SlovakBabyLM: Replication of the BabyLM and Sample-efficient Pretraining for a Low-Resource Language

Ľuboš Kriš and Marek Šuppa
Department of Applied Informatics, Comenius University in Bratislava
NaiveNeuron
Kempelen Institute of Intelligent Technologies
lubos.kris@kinit.sk, marek.suppa@fmph.uniba.sk

Abstract

In recent years, we have seen the creation of various specific language models (LMs) within the Slavic language family, which contain much fewer resources to create LMs, than other languages. However, with an increasing number of parameters of LM, a larger amount of text is required for good performance, which can hinder the development and creation of LMs for low-resource languages (LRL). Our research is looking for a solution in Curriculum learning (CL) methods that can help us build better models with a lower amount of text in comparison with current LMs, which can help in better pretraining of models with LRL. Therefore, we replicate the BabyLM Challenge in the Slovak language[1][2]. Additionally, apply CL methods for finding the difference in the application of CL methods on the English and Slovak languages, and evaluate whether the CL improves the performance of LM in the Strict-small track. Our experiments show that the use of CL methods as preprocessing methods is significant for improving model performance in sentiment analysis and question answering.

1 Introduction

According to a study by Joshi et al. (2020), seven languages are at the forefront that contain a large amount of data in the online space, which enables them to build large language models (LLMs). However, Slovak is not in that group. The lack of text in the online space may lead to less variability and text selection for LM pretraining. As part of the BabyLM challenge, researchers are trying to improve the pretraining on the amount of language needed for comprehension in children under 13 years of age (Warstadt et al., 2023b). The limited availability of Slovak text motivates this research. Furthermore, humans still outperform LLMs in certain language tasks, such as linguistic generalization, with much less data needed for language acquisition. (Beinborn and Hollenstein, 2024).

Our findings reveal that while CL offers a promising paradigm for language model pretraining, its direct application for ordering textual data, based on metrics developed for English, did not yield statistically significant performance gains for Slovak. This suggests a crucial divergence in how CL principles manifest across languages with differing morphological complexities.

2 Background

One of the main reasons for conducting the research is the difference in complexity between Slovak and English. The different complexities of the languages are inferred from their morphology, syntax, and semantics.

Morphology: The Slovak language is an inflection-based language that also has a higher number of consonants and a higher average number of morphemes in inflection. The changing word bases and modification of words are used to express different grammatical categories. In English, the focus is on derivational morphology, where specific suffixes form the word type (-ed, -ing); on the other hand, the Slovak language relies more on inflectional morphology to convey grammatical relationships (Panocová, 2021).

Syntax: English needs more words to form grammatical categories, but on the other hand they are based on a fixed word order. A sentence in Slovak (Adam mal'uje stenu- Adam paints the wall) can be expressed by switching the object and subject, but in English, the word order changes, but also words are added (Stenu mal'uje Adam-The Wall is painted by Adam). (Newerkla, 2010).

[1]Dataset: https://huggingface.co/datasets/ubokri/SlovakBabyLM.

[2]Code: https://github.com/baucek/Slovakbabylm/tree/main

Proceedings of the First BabyLM Workshop, pages 313–324

Semantics: A comparison of Slovak and English words from Swadesh's list (the Swadesh list is a list of provisionally universal terms (Swadesh, 1955)) showed that out of the 346 words examined, a total of 66 Slovak words and 97 English words had more meanings. Again, the authors attribute these results to the analytical nature and the plurality of English. That is, the English language uses more of the same words in different contexts (Urbániková, 2010).

The consequences of different complexities can be seen in the tokenization of text. A study examining text tokenization in 108 languages found that the GPT-3.5 model required 2.13 times more tokens to tokenize Slovak text than it did to tokenize English text (Asprovska and Hunter, 2024). Therefore, we consider the differences between English and Slovak to be key in applying solutions to the Babylm problem and strive to utilize them in the application of CL methods.

3 Related work

Human learning is based on the gradual acquisition of knowledge (from simple to complex) that is necessary for the development of skills needed for life (Skinner, 1958; Piaget, 2000). CL methods apply human learning abilities to LM by training them in a predetermined sequence of data based on their complexity (Bengio et al., 2009). This involves selecting rules that determine the order of training data. Therefore, metrics for measuring text complexity have been developed, which are grouped into a linguistically oriented group and a frequency-oriented group previously tested in English language (Edman and Bylinina, 2023; Bunzeck and Zarrieß, 2023; Warstadt et al., 2023b).

In the English language, linguistic complexity metrics didn't cause improvement of models. A study focusing on morphological and lexical complexity of words (Type/Token Ratio, Punctuation density, Mean word length, mean and max rarity of words) proved to be unreliable for overcoming the random ordering of text (Edman and Bylinina, 2023). Another study by (Bunzeck and Zarrieß, 2023) employed similar complexity metrics to rank text, including average word length, utterance length (the count of lexical tokens in the sequence), and average word frequency. These proved to be unsuccessful against random text order.

In frequency complexity metrics, a study retrained GPT-2 in the BabyLM dataset, using the average frequency of sentences and semantic similarity to rank and remove text, thus eliminating weak sentences of high frequency but semantically, which improved the performance of the BLIMP task (Borazjanizadeh, 2023). Another study first sorted text by complexity (spoken = "easier", typed = "harder"). Then, metrics such as the frequency of the BPE token were applied to the ranking, but this approach did not improve the BLIMP results (Martinez et al., 2023).

As mentioned in the chapter on the differences between English and Slovak, Slovak is a more semantically, morphologically, and syntactically complex language than English. From which we can assume that the metrics of linguistic complexity or frequency complexity may have a greater impact on the ranking of text in Slovak.

4 Creation of dataset

For the experiments, we had to create several sub-datasets, namely, 6 specific sub-datasets that focus on different parts of the language. Several different methods of data mining and preprocessing were used. Section Data mining and Preprocessing contains basic preprocessing, which was applied to all sub-datasets. However, due to the individual differences of the sub-datasets, each sub-dataset has a subsection for its own preprocessing.

4.1 Data mining and Preprocessing sub-datasets

For data mining the Python programming language (version 3.11.6) was used, and the following libraries were used: Scrapy (version: 2.11.1)[3], beautifulsoup (version 4.12.3)[4], requests (version: 2.32.3)[5], and the regex library (version 2023.10.3)[6]. However, each sub-dataset and source of data had a different way of using each library. The Google search engine was used to mine individual web pages containing specific data (a sub-dataset of fairy tales). In some cases, we did not find a single source with a large enough amount of data; therefore, text generation was used and the OpenAI library (version 1.35.10)[7].

The base preprocessing procedure was taken from the creation of SlovakBERT (Pikuliak et al., 2021):

[3] https://www.scrapy.org/
[4] https://pypi.org/project/beautifulsoup4/
[5] https://pypi.org/project/requests/
[6] https://docs.python.org/3/library/re.html
[7] https://openai.com/

- URL and email addresses were replaced with special tokens
- Elongated punctuation, newline character, and whitespaces were reduced, i.e., if there were sequences of the same punctuation mark, these were reduced to one mark (e.g., — to -).
- The whole text was removed if the text contains signs of wrong format
- The text is also removed if it is in a different language (using the Python library langdetect (version: 1.0.9)[8])

4.2 Sub-dataset of articles

For this sub-dataset, the main source of data was https://sk.wikipedia.org/ and the Scrapy library for web-crawling. Wikipedia contains additional links to the term that appears in the text. Thanks to this structure, subpages describing different subjects taught in primary schools were obtained from topics that a child may encounter during the teaching process (can be found in Appendix A). We ran 10 crawlers (9 pages individually as an initial_page and then 10 times all pages together).

The scrapy spider was forbidden to crawl webpages with the string in url: ("Zoznam:", "Kategória:", "Kategórie:", "Diskusia:") to remove subpages containing different lists. We also excluded the subpage "Hlavná_stránka" so that the spider did not go to the main Wikipedia page and therefore did not scrape the same content several times. In addition to the main preprocessing, we made further changes to the dataset. For incoherent scraped sources, sources with fewer than 55 characters were excluded. We also deleted sentences with specific strings "Zdroj:" or "FILIT", because they often contained inconsistent text. The final count of the text obtained can be seen in Table 1.

4.3 Sub-dataset of dialogues

We used the website https://www.opensubtitles.org/sk to create this sub-dataset. AWS cloud services, namely Lambda and S3 bucket, were used for data mining (Amazon Web Services, 2025a,b). For simplicity of the webpage, a for loop over and web-scraping with beautifulsoup4 were applied. The last code execution was on 2024-07-26. On that date, there were 29 852 movies and series with Slovak subtitles. After removing copies, incorrectly processed files, subtitles in a foreign language, applying preprocessing and non-subtitle-related sentences (translator's name, advertisement, etc.), we were left with 23 789 text files with 888 313 765 words. Due to the large amount of data, we took files that are larger than half of the total subtitle file to fit into the required number of words. The final count can be seen in the table 5.

4.4 Sub-dataset of literature

We used 4 websites containing freely available books in the Slovak language (Appendix A). The resources of https://eknizky.sk/ and https://www.1000knih.sk/ were not available for download through our web crawling or web scraping tools, so we chose manual selection, randomly selecting books that were in Slovak and were not poetry. Before preprocessing, we removed duplicate books due to their common names. Applied common preprocessing, then removed lines that contained information about the book, such as author name, publisher, ISBN, and EAN. Non-suitable pages, which contain 4 or more dots in a row (table of content) and the first page of the book, due to irrelevant text to our pretraining. With the library pymupdf library (version 1.25.5)[9], we deleted the footer and header based on the different position in comparison with other paragraphs. The final count of obtained text can be seen in Table 4.

4.5 Sub-dataset of fairytales

In creating the sub-dataset with fairytales, we used 7 webpages and a random Google search dealing with well-known fairytale authors (Appendix A). The process of creating this sub-dataset was very versatile because we worked individually with each source. Also, due to the lack of specific text, we chose the text generation method. To create the fairytales, we used the GPT-4o language model and OpenAI Library. We created two prompts to generate the fairytales (Appendix B). The first prompt was used to create the topic of the fairytale, and the second prompt was used to create the fairytale itself. A similar method of generating child-centered text can be found in other studies (Valentini et al., 2023; Schepens et al., 2023). However, we consider as a huge negative to achieve a cognitive plausible text. We cannot compare the accuracy of the created text with the real vocabulary of children at a given age,

[8] https://pypi.org/project/langdetect/

[9] https://pymupdf.readthedocs.io/en/latest/

because there is no dataset containing children's vocabulary in Slovak. The final count of obtained text can be seen in Table 2.

4.6 Sub-dataset of educational content

For the creation of the sub-dataset with educational content, the website `https://referaty.aktuality.sk/` was the main source, and `beautifulsoup4` as a tool for web-scraping due to the limited number of webpages containing content. Another reason was the exclusion of foreign languages `https://referaty.aktuality.sk/cudzie-jazyky`, which could contain a mix of foreign languages and Slovak. Besides common preprocessing, we delete sentences with ISBN, or 'Použitá literatúra', which describe sources with non-relevant text. Due to the high word count, we discarded sources with fewer than 300 characters to reduce the number of possible badly scraped data and to reduce the number of sources. Final gained text can be seen in the Table 4.

4.7 Sub-dataset of children communication

Datasets of children's communication already exist in english: the Child Language Data Ex- change System (CHILDES) (MacWhinney, 1998). No such dataset has been created within the Slovak language, and we are not aware of a free existing data source[10]. Therefore, we decided to mechanically generate conversations between a child and a known person using LLM. We used the gpt-4o language model to generate the conversations.

A similar set of mechanically generated conversations in English has already been created. The dataset of variations contains only the mother's site of conversation, and an LLM is used to repeat and rephrase, change words or their order from the inserted sentence used in the conversation (Haga et al., 2024). However, to better focus on the cognitive aspect of the conversation between the familiar person (FP) and the child, we decided to include features of the conversation between the FP and the child (We refer to the FP as a person in the child's close social circle, with whom interaction occurs, such as a parent, sibling, or guardian). According to the Usage-Based Theory of Language Acquisition (Tomasello, 1992). Communication must take place in an activity or game, where the child first passively and then actively participates. In a given game or interaction with a child, a familiar person assists in the proper development of language by repeating mispronounced words, describing the environment, or basically interacting with the child (Rowe and Snow, 2020).

Therefore, we decided to create 2 prompts (Appendix B). The first prompt creates a situation in which a child interacts with the FP, and a conversation takes place between them, which will serve as a basis for creating a conversation. The second prompt adopted the given topic and other relevant information such as the average number of words spoken by the child during a specific age, of the child. Subsequently, during prompt engineering, the greeting was randomly removed to reduce repetitive greetings. Our resulting dataset consisted of 4 groups of conversations that were created based on the settings of the child's age and the number of words used in a sentence by the child at that age.

The created text was saved as a .json file. The first prompt's output was stored under the key 'url', and the second prompt's output under 'page' to keep it consistent with other datasets. Since only the FP part of the conversation is used for pretraining, we didn't worry about the child's language accuracy and removed all lines starting with 'D:'.

[10]In other Slavic languages, it has been created `https://talkbank.org/childes/access/Slavic/`

Webpages	Number webpages	Number of words
History	16 616	6 483 417
Music	12 570	4 835 216
Chemistry	8 552	3 219 831
Sport	6 588	2 763 223
Slovak language	5 281	2 170 375
Biology	3 414	1 396 678
Physics	1 741	723 209
Civics	866	688 245
Full Subjects	93 649	35 175 212
Total	**149 277**	**57 455 406**

Table 1: Overview of scraped Wikipedia pages

Sources	Number of sources	Number of words
sikovnamamina.sk	36	41 392
rozpravkozem.sk	697	303 295
zones.sk	1 058	1 359 908
rozpravky.online	87	43 636
readmio.com/sk	1 591	359 510
svetrozpravok.sk	70	38 773
zlatyfond.sme.sk	293	671 388
Downloaded books	30	509 365
Created fairytales	3 094	1 786 974
Total	**6 956**	**4 754 731**

Table 2: Overview of fairytale sources, number of items, and word counts

Age	Number of conversations	Number of words
2 years old	9 191	478 920
3 years old	7 764	477 217
4 years old	5 433	361 184
5 years old	7 688	415 297
Total	**30 076**	**1 732 618**

Table 3: Number of conversations and words in conversations by age group

Web pages	Number of books	Number of words
zones.sk	6	246 362
eknizky.sk	210	5 628 366
greenie.elist.sk	22	423 527
1000knih.sk	44	1 383 083
Total	**272**	**7 681 338**

Table 4: Overview of scraped book sources

Domain of Sub-Dataset	Strict (Words)	Sources	Strict-small (Words)
Child-directed speech	1.7 mil	Text generation 7 webpages	470 000
Fairytales	4.7 mil	Random books Text generation	910 000
Dialogues	53.6 mil	opensubtitles.org/sk	4 000 000
Educational content	14.9 mil	referaty.aktuality.sk	1 304 000
Wiki	22 mil	sk.wikipedia.org	2 300 000
Books	7.6 mil	4 webpages	990 000
Total	**104.5 mil**		**9 974 000**

Table 5: Overview of sub-dataset domains and number of words

5 Methods

5.1 Curricullum learning criteria

The CL metrics application will consist of some of the metrics in the English versions that have already been created in the BabyLM call, plus new metrics will be created that are related to the Slovak language. The evaluation was done within a single source (downloaded book or webpage) to not divide the context of the sentences.

Linguistic complexity:

Average word length: is calculated as the number of characters divided by the number of words in a given source. This metric indicates the lexical intensity of the resource, where longer words can reduce the readability of the text.

Syllable/word ratio: is calculated as the number of syllables divided by the number of words. A given metric indicates the morphological complexity of the source, where a higher proportion of syllables per word may indicate a more complex word structure of the text.

Punctuation density: is calculated as the number of punctuation marks divided by the number of words. A low number of punctuation marks and a high number of words can mean a complex sentence structure.

Conjunction ratio: is calculated as the number of conjunctions divided by the number of words. We used 59 non-bending one-word conjunctions (Appendix C). Conjunctions link sentence constructions, which can create large sentence constructions and thus increase the syntactic complexity of the sentence (Dvonč et al., 1966).

Preposition ratio: is calculated as the number of prepositions divided by the number of words. We used 44 initial prepositions (Appendix C). Prepositions have the task of forming relations with flexible word types such as nouns or adjectives. However, at the same time, prepositions determine the case of the word they stand in front of (Dvonč et al., 1966). Inflected nouns may contain more tokens than nouns in the base form, so their presence may increase the morphological complexity of words (Asprovska and Hunter, 2024).

Frequency complexity:

Prior to actual data sorting, we extracted the frequencies of individual tokens, words, and bi-grams by splitting the words using the .split() function and removing non-alphabetical signs where appropriate.

Average word frequency: is calculated as the average of the individual word frequencies divided by the number of words in a given resource.

Average token frequency: is calculated as the average of the individual token frequencies divided by the number of words in a given resource.

Average bi-gram frequency: is calculated as the average of the individual bigram frequencies divided by the number of words in a given source.

In order to properly measure the given metrics (lower rankings == simpler sentences), we had to rescale the frequency group metrics and punctuation density metric (metric=-1*metric). The metrics were normalized using min-max normalization across the entire dataset. The order is defined as the sum of normalized values. If the experiment required the ranking of sub-datasets, the points were sum for each sub-dataset, and the ranking of sub-datasets within a single metric was determined (1 = simplest according to the metric and 6 = most complex according to the metric).

5.2 Architecture and Pretraining

The architecture of the models was based on the results of the BabyLM challenge (Warstadt et al., 2023b). Hyper-optimization of the hyperparameters prove a 1:2 ratio between feed-forward layers and attention heads as the best (Proskurina et al., 2023). Therefore, our models had 6 layers and 12 attention heads. For training, we used a sequence length of 128 tokens and a batch size of 128, both identified as optimal parameters in prior research (Cagatan, 2023; Proskurina et al., 2023). We applied a 15% masking rate across 7 training epochs, following established best practices (Cagatan, 2023). The given studies used a Vocabulary size of 40000 and 30000 (Edman and Bylinina, 2023; Opper et al., 2023), but to handle Slovak's complex morphology, we set a larger vocabulary of 60,000 tokens with byte-level BPE tokenization. To pre-train and test models, we used two graphics cards NVIDIA GeForce RTX 3090 and NVIDIA GeForce GTX 1080.

5.3 Testing parameters

Within the Babylm challenge (Warstadt et al., 2023a), researchers used several evaluation tasks. Hence, we will use sentiment analysis (SA), question answering (QA) tasks. The SA task focuses on identifying specific word combinations. The dataset `dgurgurov/slovak_sa` (Pecar et al.,

2019) was selected for the SA task and 4 performance measurements: Accuracy, Precision, Recall, and F1-score. The QA task tests the model's ability to process and generate answers by analyzing word relationships (Farea et al., 2022). Dataset `TUKE-DeutscheTelekom/squad` (Hládek et al., 2023) was selected, and the F1 score and exact match were selected as performance measurement. The models were fine-tuned for specific tasks, and each model was tested 9 times with 2 different hyperparameters (3 different learning rates[11] and 3 different epochs[12]) and then performed statistical significance testing to evaluate performance differences between model variations and the model without improvements. T-test was used for each of the selected evaluation metrics.

6 Results

For the purpose of testing CL metrics in practice, 7 LMs were created with Strict-small dataset. 5 LMs can be divided into two groups according to the criteria: sorting and specific CL metric groups. The other 2 LMs had selected data from strict track into strict-small track based on complexity.

Application of specific ordering:

1. Without ordering, without specific group metrics
2. Sub dataset ordering, both metric groups
3. Full ordering, both metric groups

Application of group metrics:

4. Full ordering, only language group metric
5. Full ordering, only frequency group metric

6.1 Application of CL methods

From the results in Tables 6 and 7, we can conclude that no group of metrics is significant for model improvement. From average of all trials (Appendix D.1), we can observe better performance of the linguistic group against the frequency group. This may indicate a potentially higher relevance of language-based features versus frequency-based features. It can be interpreted as the richness of the Slovak language, where a larger number of tokens is created. On the other hand, Individual linguistic or frequency groups of metrics can influence LM differently. Specifically, CL improvement in the SA task is based on the gradual increase in the variation of nouns and verbs as significant factors (Elgaar and Amiri, 2023).

Comparison	t-value	p-value
3. Both groups of metrics		
Accuracy	-1.59	0.924
Recall	-1.59	0.924
F1	-1.39	0.899
Precision	-1.38	0.897
4. Language group		
Accuracy	-1.00	0.827
Recall	-1.00	0.827
F1	-0.47	0.675
Precision	-0.16	0.562
5. Frequency Group		
Accuracy	-1.70	0.936
Recall	-1.70	0.936
F1	-1.35	0.894
Precision	-1.25	0.876

Table 6: Statistical significance (p>= 0.05) of models using CL methods compared to the model without CL methods in SA task

Comparison	t-value	p-value
3. Sum + Frequency (grammar)		
Exact Match	-0.53	0.694
F1	-1.24	0.875
4. Grammar Group		
Exact Match	0.04	0.484
F1	-0.54	0.699
5. Frequency Group		
Exact Match	-2.93	0.990
F1	-1.18	0.864

Table 7: Statistical significance (p>= 0.05) of models with CL methods and the model without CL methods in QA task

6.2 Text ordering methods

Models with different text ordering did not prove to be significantly better than the model without any application of the CL metrics and spcific ordering (Tables 8 and 9). The ordering of the sorted sub-datasets shows worse performance in QA performance than the model with the ordering of full data based on F1 score and exact match, and in turn, the ordering of full data performs worse in SA tasks based on F1 score, precision, accuracy, and loss (Appendix D.2). Results suggest the possibility that data ordering may affect context handling performance. In Malkin et al. (2021) demonstrate the absence of coherence and logic as a negative factor to handle longer-term dependencies between sentences and effective context work. As for SA tasks, this can be explained by the effect of variations in nouns and verbs, which by using metrics and ranking the whole dataset can effect this ranking (Elgaar and Amiri, 2023).

[11] learning rate= [5e-5,3e-5,1e-5]
[12] epochs = [5,7,10]

Comparison	t-value	p-value
2. ordering of sub-datasets		
Exact Match	-0.05	0.518
F1	-0.50	0.684
3. ordering full data		
Exact Match	-0.53	0.694
F1	-1.24	0.875

Table 8: Statistical significance (p>= 0.05) of models with specific application of CL methods and the model without CL methods in QA task

Metric	t-value	p-value
2. ordering of sub-datasets		
Accuracy	-2.54	0.983
Recall	-2.54	0.983
F1 Score	-2.28	0.974
Precision	-2.39	0.978
3. ordering full data		
Accuracy	-1.59	0.924
Recall	-1.59	0.924
F1 Score	-1.39	0.899
Precision	-1.38	0.897

Table 9: Statistical significance (p>= 0.05) of models with specific application of CL methods and the model without CL methods in SA task

6.3 Metrics as preprocessing methods

The following results show that using CL metrics for preprocessing has the highest effect among the applied improvements. The application of the hardest complexity on the QA task show significant improvement by F1 score (Table 10) and the simplest texts for pretraining the model on the SA task (Table 11). After scanning the sources used for pretraining, we can infer that resources contain long words or disjointed text, which can be reduced by selecting the simplest sources. Therefore model pretrained on text with the hardest complexity was better in QA task, and the model pretrained on text with the simplest complexity was better in the SA task. This again confirms the relationship between specific CL methods and performance improvement in specific tasks. (Elgaar and Amiri, 2023).

Metric	t-value	p-value
1. The simplest complexity		
Exact Match	-2.35	0.976
F1	-4.48	0.684
1. The hardest complexity		
Exact Match	0.76	0.233
F1	2.19	*0.030

Table 10: Statistical significance (p>= 0.05) of models with selection of the text based complexity and the model without any improvements in the QA task

Metric	t-value	p-value
1. The simplest complexity		
Accuracy	1.57	0.077
Recall	1.57	0.077
F1 Score	1.79	0.056
Precision	1.81	*0.054
1. The hardest complexity		
Accuracy	0.78	0.228
Recall	0.78	0.228
F1 Score	0.12	0.454
Precision	-0.08	0.532

Table 11: Statistical significance (p >= 0.05) of models with selection of the text based complexity and the model without any improvements in the SA task

7 Conclusion

The aim is to establish a cornerstone in the research of cognitively inspired models in the Slovak language and to point out the possibilities of applying CL to LRLs such as the Slovak language. In pursuit of this goal, a Slovak version of the BabyLM challenge (Warstadt et al., 2023a) was created. The constructed experiments demonstrated several findings. They confirmed the results of studies in English, where both sets of metrics showed no significant improvement in QA and SA tasks (Martinez et al., 2023; Bunzeck and Zarrieß, 2023; Edman and Bylinina, 2023).

CL metrics as preprocessing methods shows significant improvement against random order of text. These results can lead to use linguistic and frequency CL metrics as a potential optimization text. Similar usage of CL can be seen in data selection for abstractive text summarization (Sun et al., 2023). Applied CL identify high-value training examples, demonstrating that targeted data selection can improve model performance compared to training on the full dataset. The subtle differences noted, often visible only in decimal places (Appendix D), emphasize the challenge of discerning the impact of CL strategies when baseline performance is already competitive or when the dataset size limits the magnitude of observable improvements.

These nuances could be indicative of slight shifts in the model's learned representations, even if not leading to a statistically superior overall score. Additionally, the positive or negative effect of CL metrics in Slovak was much less significant than in English. According to (Bengio et al., 2009), CL metrics need to gradually increase the amount of more useful information with increasing learning time, which may be more complicated in Slovak language due to its linguistic complexity, where

the order produced by the metrics may be more difficult to determine compared to English. The greater semantic complexity of the Slovak language can lead to a higher number of tokens (Asprovska and Hunter, 2024). This factor can lead to subtle frequency variations that may not provide sufficiently clear signals for learning within a robust curriculum. What could have led to the failure of frequency methods.

Limitations

One of the fundamental limitations of the created dataset and results was the mechanical generation of data. This is not valid from the perspective of adapting to human learning. At the same time, LRLs may also lack dictionaries of children's speech, such as the Slovak language, which can serve as a test of the generated data or an aid to their creation (Schepens et al., 2023; Haga et al., 2024) or verification for the developmentally plausible of created text. Thus, it is not only a limitation but also a call to action for researchers to invest in improving Slovak language resources and model support, and also research in multilingual research in cognitive-inspired language models.

Within the limitations of computational resources, we could not focus on the semantic or syntactic part of the language. The results of studies with cosine similarity as an evaluator for CL metrics show that the factor would improve the evaluation, and hence we would get better results (Han and Myaeng, 2017; Borazjanizadeh, 2023)). For instance, due to the computational complexity of metrics, we could not perform the Part of Speech evaluation or more deeper analysis of text.

Acknowledgements

This work has been partially supported by grant APVV-21-0114. Additionally, we would like to thank our affiliations Kempelen Institute of Intelligent Technologies for providing technical advice, and the Faculty of Mathematics, Physics, and Informatics, for providing the computing resources used to pre-train our language models.

References

Amazon Web Services. 2025a. Amazon S3.

Amazon Web Services. 2025b. AWS Lambda. Accessed: 2025-04-12.

Marijana Asprovska and Nathan Hunter. 2024. The tokenization problem: Understanding generative ai's computational language bias. *Ubiquity Proceedings*, 4(1).

Lisa Beinborn and Nora Hollenstein. 2024. *Cognitive plausibility in natural language processing*. Springer.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Nasim Borazjanizadeh. 2023. Optimizing gpt-2 pretraining on babylm corpus with difficulty-based sentence reordering. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 356–365.

Bastian Bunzeck and Sina Zarrieß. 2023. Gpt-wee: How small can a small language model really get? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 35–46.

Omer Veysel Cagatan. 2023. Toddlerberta: Exploiting babyberta for grammar learning and language understanding. *arXiv preprint arXiv:2308.16336*.

Ladislav Dvonč, Jozef Ružička, et al. 1966. Morfológia slovenského jazyka. *(No Title)*.

Lukas Edman and Lisa Bylinina. 2023. Too much information: Keeping training simple for babylms. *arXiv preprint arXiv:2311.01955*.

Mohamed Elgaar and Hadi Amiri. 2023. Ling-cl: Understanding nlp models through linguistic curricula. *arXiv preprint arXiv:2310.20121*.

Amer Farea, Zhen Yang, Kien Duong, Nadeesha Perera, and Frank Emmert-Streib. 2022. Evaluation of question answering systems: complexity of judging a natural language. *arXiv preprint arXiv:2209.12617*.

Akari Haga, Akiyo Fukatsu, Miyu Oba, Arianna Bisazza, and Yohei Oseki. 2024. Babylm challenge: Exploring the effect of variation sets on language model training efficiency. *arXiv preprint arXiv:2411.09587*.

Sanggyu Han and Sung-Hyon Myaeng. 2017. Tree-structured curriculum learning based on semantic similarity of text. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 971–976. IEEE.

Daniel Hládek, Ján Staš, Jozef Juhár, and Tomáš Koctúr. 2023. Slovak dataset for multilingual question answering. *IEEE Access*, 11:32869–32881.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. In *Proceedings of the 58th Annual Meeting of*

the Association for Computational Linguistics, pages 6282–6293. Association for Computational Linguistics.

Brian MacWhinney. 1998. The childes system. *Handbook of child language acquisition*, pages 457–494.

Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. 2021. Coherence boosting: When your pretrained language model is not paying enough attention. *arXiv preprint arXiv:2110.08294*.

Richard Diehl Martinez, Zebulon Goriely, Hope McGovern, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. Climb: Curriculum learning for infant-inspired model building. *arXiv preprint arXiv:2311.08886*.

Stefan-Michael Newerkla. 2010. Juraj dolník, všeobecná jazykoveda. opis a vysvetl'ovanie jazyka, bratislava (veda, vydavatel'stvo slovenskej akadémie vied) 2009, 376 s.

Mattia Opper, J Morrison, and N Siddharth. 2023. On the effect of curriculum learning with developmental data for grammar acquisition. *arXiv preprint arXiv:2311.00128*.

Renáta Panocová. 2021. Basic concepts of morphology i. *Košice: Vydavatel'stvo ŠafárikPress*.

Samuel Pecar, Marián Šimko, and Maria Bielikova. 2019. Improving sentiment classification in slovak language. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 114–119.

Jean Piaget. 2000. Piaget's theory of cognitive development. *Childhood cognitive development: The essential readings*, 2(7):33–47.

Matúš Pikuliak, Štefan Grivalskỳ, Martin Konôpka, Miroslav Blšták, Martin Tamajka, Viktor Bachratỳ, Marián Šimko, Pavol Balážik, Michal Trnka, and Filip Uhlárik. 2021. Slovakbert: Slovak masked language model. *arXiv preprint arXiv:2109.15254*.

Irina Proskurina, Guillaume Metzler, and Julien Velcin. 2023. Mini minds: Exploring bebeshka and zlata baby models. *arXiv preprint arXiv:2311.03216*.

Meredith L Rowe and Catherine E Snow. 2020. Analyzing input quality along three dimensions: interactive, linguistic, and conceptual. *Journal of child language*, 47(1).

Job Schepens, Nicole Marx, and Benjamin Gagl. 2023. Can we utilize large language models (llms) to generate useful linguistic corpora? a case study of the word frequency effect in young german readers. *Preprint from PsyArXiv https://doi. org/10.31234/osf. io/gm9b6*.

Burrhus F Skinner. 1958. Reinforcement today. *American Psychologist*, 13(3):94.

Shichao Sun, Ruifeng Yuan, Jianfei He, Ziqiang Cao, Wenjie Li, and Xiaohua Jia. 2023. Data selection curriculum for abstractive text summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7990–7995.

Morris Swadesh. 1955. Towards greater accuracy in lexicostatistic dating. *International journal of American linguistics*, 21(2):121–137.

Michael Tomasello. 1992. *First verbs: A case study of early grammatical development*. Cambridge University Press.

Milica Urbániková. 2010. Lexical and semantic development of the basic vocabulary in english and slovak.

Maria Valentini, Jennifer Weber, Jesus Salcido, Téa Wright, Eliana Colunga, and Katharina Kann. 2023. On the automatic generation and simplification of children's stories. *arXiv preprint arXiv:2310.18502*.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for papers–the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2301.11796*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. 2023b. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*.

A Appendix: Web-pages for sub-datasets

Sub-dataset of articles: https://sk.wikipedia.org/wiki/ObÄĎianska_nÃąuka,https://sk.wikipedia.org/wiki/Dejiny, https://sk.wikipedia.org/wiki/Sloven%C4%8Dina, https://sk.wikipedia.org/wiki/Fyzika, https://sk.wikipedia.org/wiki/Matematika, https://sk.wikipedia.org/wiki/Informatika, https://sk.wikipedia.org/wiki/Ch%C3%A9mia, https://sk.wikipedia.org/wiki/Biol%C3%B3gia, https://sk.wikipedia.org/wiki/Hudba, https://sk.wikipedia.org/wiki/%C5%A0port Sub-dataset of literature: https://www.zones.sk/, https://eknizky.sk/, https://greenie.elist.sk/, https://www.1000knih.sk/ Sub-dataset of fairytales: https://www.sikovnamamina.sk/, https://www.rozpravkozem.sk/, https://www.zones.sk/studentske-prace/rozpravky/, https://rozpravky.online/, https://zlatyfond.sme.sk/, https:

//www.readmio.com/sk/uvod, https://svetrozpravok.sk/

B Appendix: Prompts for text generation

Children's books:
First prompt:
count = 200
"role": "user", "content": f"Vytvor mi {count} názvov rozprávok. Vráť len zoznam názvov rozprávok bez ďalšieho úvodu, číslovania alebo záverečných poznámok. Témy sa nesmú opakovať.", "role": "assistant", "content": "Si spisovateľ detských rozprávok."

Second promt: "role": "user", "content": f"Vytvor rozprávku pre deti na tému:{topic}. Snaž sa využiť maximálny počet tokenov.", "role": "assistant", "content": "Si spisovateľ detských rozprávok."

Child-directed speech:
count = 200
family = 'Matkou'
age_word = 'dvojročným'
age = 2
average_number_words = 2
First prompt: "role": "user", "content": f"""Vytvor {count} situácii medzi {family} a dieťaťom, ktoré sa môžu vyskytnúť medzi {family } a {age_word } dieťaťom. Výsledok budú len dané situácie a nebudú sa opakovať. Príklad: Prebaľovanie. Dieťa nesmie byť súčasťou činnosti, ktorú je nemožné vykonať v danom roku života ({age } roky). """, "role": "assistant", "content": "Si schopný posúdiť lingvistické a motorické prejavy dieťaťa v rôznom veku."
Second prompt: "role": "user", "content": F"""Daj mi konverzáciu medzi {age_word} dieťaťom a {family } na tému:{topic}. Tvoj výsledok musí obsahovať len vytvorený dialóg, kde {family } bude označená ako {family[0].upper() }: a dieťa ako D. Správna komunikácia zo strany {family}: Komentovanie: Je dôležité opisovať, to čo sa deje v okolí. Opakovanie: Zdôrazňovať a opakovať veci, ktorým dieťa nerozumie a poskytnúť možností na neustále opakovanie nových slov alebo viet Výslovnosť: Použi slová gramaticky správne. !!!!! Prispôsobivosť: family musi prispôsobiť reč aktuálnym záujmom a potrebám dieťaťa: vety sú krátke!!!!! Priemerný počet slov vo vete u dieťaťa: {average_number_words}. Nezačínaj komunkáciu pozdravom""", "role": "assistant", "content": "Si schopný posúdiť lingvistické a motorické prejavy dieťaťa v rôznom veku."

C Appendix: List of conjunctions and prepositions

Conjunctions = (a, že, i, keby, aby, aj, ak, keď, keďže, ako, akoby, hoci, ale, alebo, lebo, ani, iba, tak, takže, teda, totižto, veď, však, žeby avšak, až, ba, bár, beztak, buď, by, či, čím, čoby, pričom čiže, čo, kým, leda, ledva, len, len čo, totiž, lenže, najprv, nech, než, nielen, no, nuž, pokiaľ, pokým, predsa, preto, pretože, síce, sotva, sťa, prí padne, poprípade, eventuálne)

Prepositions = (bez, cez, do, k, medzi, na, nad, o, od, okrem, po, pod, pre, pred, pri, proti, s, skrz, u, v, z, za, ponad, popod, popred, poza, popri, pomedzi, znad, spred, zmedzi, spod, spopred, sponad, spopod, spoza, spopri, spomedzi, zo, ku, voči, skrze, vo, so)

D Appendix: Mean of results

D.1 Application of CL methods

Models	Exact Match (%)	F1 Score (%)
1. Without ord.	1.39 (0.38)	**6.59 (0.65)**
3. Both groups	1.33 (0.29)	6.44 (0.75)
4. Language group	**1.39 (0.37)**	6.45 (1.20)
5. Frequency group	1.22 (0.37)	6.45 (0.74)

Table 12: Results of QA task for Text ordering methods (mean ± std).

Metric	1. Without ord.	3. Both groups	4. Lang. group	5. Freq. group
Loss	**0.32 (0.11)**	0.32 (0.11)	0.32 (0.10)	0.34 (0.10)
Acc (%)	**90.38 (2.65)**	90.16 (2.53)	90.23 (2.59)	89.66 (2.34)
Prec (%)	**85.45 (8.13)**	85.14 (7.90)	85.41 (8.10)	83.51 (7.81)
Rec (%)	**90.38 (2.65)**	90.16 (2.53)	90.23 (2.59)	89.66 (2.34)
F1 (%)	**87.74 (5.54)**	87.43 (5.34)	87.66 (5.49)	86.36 (5.23)

Table 13: Results of SA task for application of CL metrics (mean ± std).

D.2 Text ordering methods

Models	Exact Match (%)	F1 Score (%)
1. without ordering	**1.38 (0.38)**	**6.59 (0.65)**
2. ordering of sub-datasets	1.38 (0.41)	6.54 (0.60)
3. ordering full data	1.33 (0.29)	6.44 (0.75)

Table 14: Results of QA task for Application of CL methods: (mean ± std).

Metric	1. Without ord.	2. Sub-datasets	3. Full data
Loss	**0.32 (0.11)**	0.34 (0.10)	0.32 (0.11)
Acc (%)	**90.38 (2.65)**	89.74 (2.18)	90.16 (2.53)
Prec (%)	**85.45 (8.13)**	84.56 (7.36)	85.14 (7.90)
Rec (%)	**90.38 (2.65)**	89.74 (2.18)	90.16 (2.53)
F1 (%)	**87.74 (5.54)**	86.88 (4.85)	87.43 (5.34)

Table 15: Results of SA task for Application of CL methods: (mean ± std).

D.3 Metrics as preprocessing methods

Models	Exact Match (%)	F1 Score (%)
1. Random complexity	1.39 (0.38)	6.59 (0.65)
1. The simplest complexity	**1.60 (0.25)**	**6.95 (0.70)**
1. The hardest complexity	1.28 (0.36)	6.07 (1.25)

Table 16: Results of QA task for metrics as preprocessing methods (mean ± std).

Metric	1. Random	1. The simplest	1. The hardest
Loss	**0.32 (0.11)**	0.33 (0.10)	0.32 (0.10)
Acc (%)	**90.38 (2.65)**	90.05 (2.43)	90.32 (2.61)
Prec (%)	85.45 (8.13)	85.05 (7.76)	**85.46 (8.13)**
Rec (%)	**90.38 (2.65)**	90.05 (2.43)	90.32 (2.61)
F1 (%)	**87.74 (5.54)**	87.36 (5.20)	87.73 (5.53)

Table 17: Results of SA task for metrics as preprocessing methods (mean ± std).

E Appendix: Model settings for pretraining

BertConfig
vocab_size = 60000
hidden_size = 84
num_hidden_layers = 6
num_attention_heads = 12
intermediate_size = 1446
hidden_dropout_prob = 0.15
attention_probs_dropout_prob = 0.3
hidden_act = "gelu_new"
TrainingArguments
num_train_epochs = 7
per_device_train_batch_size = 32
per_device_eval_batch_size = 32
evaluation_strategy = "steps"
eval_steps = 1000
save_steps = 1000
logging_steps = 100
load_best_model_at_end = True
metric_for_best_model = "eval_loss"
bf16 = True
Finetuning of SA and QA
weight_decay=0.01,
per_device_train_batch_size=16,
per_device_eval_batch_size=16,
save_strategy="epoch",
evaluation_strategy="epoch",
Specific QA parameters
n_best = 20
max_answer_length = 50

Single layer tiny Co^4 outpaces GPT-2 and GPT-BERT

Noor Ul Zain[1], Mohsin Raza[1], Ahsan Adeel[1,*]
[1] CMI-Lab
University of Stirling, UK
[*] ahsan.adeel1@stir.ac.uk

Abstract

We show that a tiny Co^4 machine (Adeel, 2025) with a single layer, two heads, and 8M parameters, operating at an approximate cost of $O(N)$ (where N is the number of input tokens), outpaces the BabyLM Challenge baselines GPT-2[1] (124M, 12 layers, $O(N^2)$) and GPT-BERT[2] (30M, 12 layers, $O(N^2)$) in just two epochs, while both are trained for ten. Co^4 achieves orders-of-magnitude greater training efficiency on 10M tokens, demonstrating highly sample-efficient pretraining. Using the BabyLM challenge evaluation pipeline across complex benchmarks, Co^4 exhibits strong zero-shot and fine-tuning performance on SuperGLUE tasks. Specifically, Co^4 outperforms GPT-2 on 5 out of 7 zero-shot metrics and 6 out of 7 fine-tuning tasks, and GPT-BERT on 4 out of 7 metrics in both cases. These results suggest the need to rethink prevailing deep learning paradigms and associated scaling laws.

Cellular neurobiological evidence (Suzuki et al., 2023; Marvan and Phillips, 2024) on how mammalian brains achieve fast and flexible computation continues to challenge deep (hierarchical) learning (LeCun et al., 2015; Vaswani et al., 2017; Wang et al., 2025), predictive coding (Rao and Ballard, 1999; Friston, 2005, 2010), and scaling laws (Kaplan et al., 2020). Evidence suggests that the brain's computational power lies in shallow architectures, where cortical and subcortical networks operate with massive parallelism, leveraging cortical microcircuits and thalamo-cortical loops (Aru et al., 2021; Storm et al., 2024; Phillips et al., 2024) to support faster, context-sensitive, and coherent internal understanding (Adeel, 2025).

Modern deep learning architectures, such as Transformers (Vaswani et al., 2017; Jaegle et al., 2021; Alayrac et al., 2022), which underpin models like GPT and GPT-BERT, act as sequential local agents reducing predictive error or free energy (Friston, 2005, 2010), yet without regard for local coherence (Marvan and Phillips, 2024). During the feedforward (FF) phase, they lack intrinsic mechanisms to judge the true relevance of an attended token (Adeel, 2025). Instead, relevance is indirectly shaped by backpropagation during the feedback (FB) phase, a brute-force, reward-driven process. Incoherent inferences generated by initial agents (e.g., early transformer blocks) propagate to subsequent agents, where they are reinforced through ineffective FB signals. We refer to this as a "Chinese Whispers" problem.

Consequently, these deep nets require vast datasets, extensive training time, and significant compute, resulting in unsustainable economic, environmental, and technical costs (Thompson et al., 2020). The reliance on deeper architectures for hierarchical feature abstraction is a shared limitation across other neural models, including long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), gated-recurrent units (GRUs) (Chung et al., 2014), and convolution neural networks (CNNs) (LeCun et al., 1989).

The recently proposed Co^4 machine (Adeel, 2025) emulates higher-level perceptual processing (HLPP) and awake thought (AT) mental states (Phillips et al., 2024). Within a single layer, during FF, it executes triadic FB loops among latent questions (Qs), clues (Ks), and hypotheses (Vs), enabled by three two-point neurons (TPNs)[3] (Aru et al., 2021; Storm et al., 2024; Phillips et al., 2024), each representing an agent holding K, Q, and V. Unlike Transformers, which propagate layer-wise,

[1] https://huggingface.co/BabyLM-community/babylm-baseline-10m-gpt2

[2] https://huggingface.co/BabyLM-community/babylm-baseline-10m-gpt-bert-causal-focus

[3] A pyramidal two-point neuron in the mammalian neocortex integrates feedforward input at its basal site and contextual input at its apical dendrites. When both are aligned in time, the neuron fires bursts that amplify coherent, contextually relevant signals for active inference.

Proceedings of the First BabyLM Workshop, pages 325–334

Co^4 enables all agents to co-evolve Qs, Ks, and Vs in parallel: Qs update based on Ks and Vs; Ks update based on Qs and Vs; Vs evolve based on Ks and Qs. Each TPN agent independently forms distinctive Q–K–V perspectives, thereby maximizing local and global coherence (Marvan and Phillips, 2024) while minimizing free energy (Friston, 2005, 2010), ensuring token relevance before attention is applied or decisions are made. This cooperative mechanism enables diverse, parallel, and deep reasoning chains without requiring additional layers, at an approximate cost of $O(N)$ (Adeel, 2025).

This paper is the first to report the Co^4 machine's performance on complex language benchmarks. From a cognitive modeling perspective, we compare training trajectories of Co^4, GPT-2, and GPT-BERT to those of children using psycholinguistic metrics under data-limited conditions modeled after human language acquisition (Charpentier et al., 2025). Despite its tiny size, just one layer, two heads, and 8M parameters, Co^4 (with $O(N)$ cost) outpaces GPT-2 (124M parameters) and GPT-BERT (30M), both using 12 layers ($O(N^2)$ cost), achieving orders-of-magnitude greater efficiency and stronger generalization on a 10M-token dataset.

1 Neurons and Co^4 agents with two points of input integration

Going beyond the 20^{th}-century neuroscience conception of point neurons (PNs) (Häusser, 2001), on which most current brain theories and AI systems are based, 21^{st}-century neuroscience (Larkum et al., 1999; Phillips, 2017, 2023; Larkum, 2013; Major et al., 2013; Ramaswamy and Markram, 2015; Larkum, 2022; Adeel, 2020; Körding and König, 2000; Schuman et al., 2021; Poirazi and Papoutsi, 2020; Larkum et al., 2018; Shine et al., 2016, 2019; Shine, 2019; Shine et al., 2021; Schulz et al., 2021; Kay and Phillips, 2020; Kay et al., 2022) has revealed that certain neurons, particularly some pyramidal neurons in the mammalian neocortex, integrate inputs at two distinct locations. These are often referred to as TPNs, which combine information from the external environment (feedforward (FF)) at one site (basal) and contextual (C) input at another (apical). TPNs trigger high-frequency firing (bursting) when the FF and C inputs are matched in time, that is, when both the basal and apical zones are depolarized. This results in the amplification of coherent signals, enabling enhanced contextually rich processing (Phillips et al., 2024).

The flexible interaction between FF and C inputs is suggested to be the hallmark of conscious processing (Aru et al., 2021; Storm et al., 2024; Marvan et al., 2021) and linked to distinct mental states, including wakefulness (WF), slow-wave (SW) sleep, and rapid eye movement (REM) sleep (Phillips et al., 2024). Dysfunctional interactions between FF and C inputs have been linked to intellectual learning disabilities (Nelson and Bender, 2021; Granato et al., 2024).

Several TPN-inspired machine learning algorithms have been proposed to flexibly combine top-down C and bottom-up FF information streams (Payeur et al., 2021; Greedy, 2022; Guerguiev et al., 2017; Sacramento et al., 2018; Illing et al., 2022; Greedy, 2022; Zenke et al., 2017; Kirkpatrick et al., 2017; Kastellakis et al., 2016; Bono and Clopath, 2017; Limbacher and Legenstein, 2020). However, most of these efforts have focused on using apical (contextual) inputs primarily for learning. Ample evidence suggests that the apical site not only receives feedback from higher perceptual levels but also integrates simultaneous events across multiple hierarchical levels while processing FF information. For example, results using TPN-inspired CNNs (Adeel, 2020; Adeel et al., 2022, 2023; Raza and Adeel, 2024) showed that these architectures could drastically reduce the transmission of conflicting FF signals to higher perceptual areas, achieving orders-of-magnitude reductions in the number of neurons needed to process heterogeneous real-world audio-visual data, compared to standard PN-based CNNs. More recent findings demonstrate that the TPN-inspired Co^4 machine (Adeel, 2025), emulating higher level perceptual processing and imaginative thought mental states can enable significantly faster learning with substantially lower computational demands (e.g., fewer heads, layers, and tokens) at an approximate cost of $O(N)$. These gains were observed across a variety of domains, including reinforcement learning, computer vision, and natural language question answering.

These efforts to develop efficient machine learning models align with scaled-down pretraining using fewer than 100M tokens, evaluating language models (LMs) on the same types and quantities of data that humans are exposed to (Charpentier et al., 2025). The aim is to build plausible cognitive models of human learning and to better understand how children are exposed to language with such ef-

ficiency. By combining cellular neurobiologically inspired, TPN-based Co^4 machine (Adeel, 2025) with this scaled-down pretraining strategy, we introduce the Co^4 LM.

2 Co^4 Language Model

Figure 1 (left) illustrates the standard GPT-2 model, consisting of 12 Transformer layers, where each layer performs a simple conclusion via self-attention (QK^TV) at the cost of $O(N^2)$. This can be interpreted as 12 agents working sequentially. The selection of relevant and irrelevant tokens in the FF phase is determined through backpropagation, a brute-force process solely driven by the global objective. This rigidity causes the network to depend heavily on pre-learned patterns, limiting its ability to generate new perspectives quickly. When initial thoughts are misleading, arriving at a correct conclusion may require significantly more time and computation, or may not happen at all, due to limited internal flexibility and constrained cognitive resources (Adeel, 2025).

In contrast, Figure 1 (right) shows a single-layer Co^4 machine with two attention heads. After initializing the latent queries (Qs) as a set of neuronal agents (e.g., 24) (as opposed to 12 attention blocks + feedforward neuron network (FFNN) in GPT-2 and GPT-BERT), they begins to co-evolve their own Qs, Ks, and Vs in parallel during the FF phase via triadic modulation loops leveraging proximal (P), distal (D), and universal (U) contextual fields. This co-evolution is enabled through inherent, moment-by-moment cooperation mechanisms or asynchronous modulation (MOD) transfer function (Adeel, 2025), resulting in rich, contextually-aware, and diverse parallel reasoning chains at the cellular level. Each agent independently develops its own Q, K, and V, leading to 24 attention maps and 24 possibly different conclusions. Importantly, this all occurs virtually, allowing the model to pre-select relevant tokens before applying latent self-attention at an approximate cost of $O(N)$ (Adeel, 2025).

The Co^4 language model frames text generation as an autoregressive, left-to-right process: given a prefix of tokens, the model computes a probability distribution over the next token via a softmax applied to its hidden state. We use the same tokenizer as the baselines. The input tokens are first mapped to continuous vectors through a embedding layer and are augmented with positional embeddings to encode sequence order. During training, a triangular causal mask ensures that each position can only attend to previous positions. The model's weights are optimized by minimizing the cross-entropy (CE) loss (equivalently, the negative log-likelihood) of the true next token.

The Co^4 language model condenses this pipeline into a single decoder layer with just two attention heads, yet enriches it via triadic modulation loops among Q-, K-, and V-TPNs, operating through P, D, and U contextual fields (Adeel, 2025, 2020). After token embedding and positional projection, each token's Q, K, and V vectors co-evolve through a series of rapid and modulated updates.

We trained Co^4 on a 10M-token slice of the BabyLM corpus (BabyLM Community, 2023), using the same autoregressive CE objective but at a fraction of the training budget of GPT-2 and GPT-BERT, which are the official baselines provided by the organizers of this challenge. More details related to the hyperparameters for these baselines can be found on the relevant model repositories on Hugging Face.

3 Results

In this section, we present the performance of our tiny Co^4 machine across a range of language modeling benchmarks. The seven tasks described first assess the model's linguistic capabilities in a purely zero-shot setting, without any additional training or fine-tuning. Later in the section, we also evaluate Co^4's performance on fine-tuning benchmarks and provide an extensive comparison with the baseline. We utilize the evaluation suite from the BabyLM Challenge (Charpentier et al., 2025), which includes the following zero-shot metrics. The first two, newly introduced, are designed to compare the language model's responses to those of human judgments and behavioral data.

- Eye Tracking and Self-paced Reading: This psycholinguistic measure evaluates whether the model can mimic the eye tracking and reading time of a human by using the surprisal of a word as a proxy for time spent reading a word (de Varda et al., 2024).
- WUGs: morphological Adapting the classic "Wug" paradigm, this evaluates whether models can generalize morphological rules to form novel noun derivatives from unseen adjectives, and compares the model's generalization to that of humans (Hofmann et al., 2025).

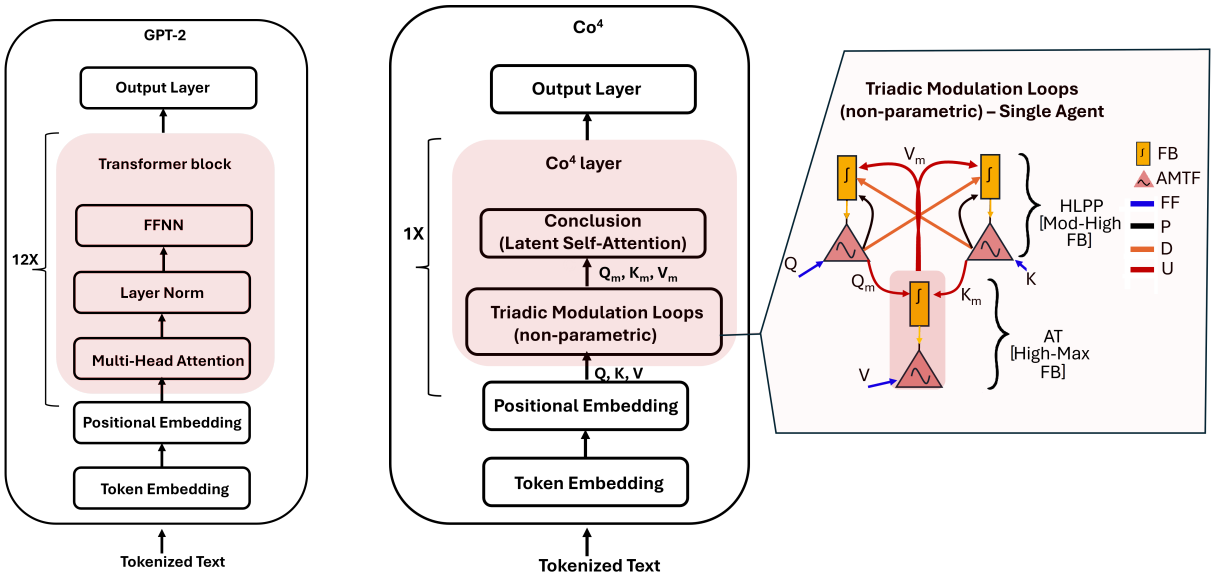


Figure 1: Language Models: GPT-2 (Left) vs. Co^4 (Right). In Co^4, the learnable parameters are only in the embedding layer and the initial Q, K, V representations, followed by a single layer of non-parametric triadic modulation loops (referred to as "1x" Co4 or single-layered Co4). Co^4 does not require feed feed-forward neural network (FFNN/ MLP) layer used in standard GPT-type architectures. Inside these loops, three populations of three pyramidal two-point processors, each associated with Q, K, and V, respectively, simultaneously integrate FF information and FB context at two functionally distinct sites. The apical (top-down) site (shown in the rectangle) integrates context, while FF information is integrated at the basal (bottom-up) site (shown in the triangle). Each processor, via asynchronous modulation (MOD) transfer functions[4], operating in higher-level perceptual processing (HLPP) or awake thought (AT) mode, depending on the strength of FB, amplifies FF transmission if it is relevant in that context (represented by P, D, U). Otherwise, it attenuates the signal, resulting in the selective amplification of coherent FF information (Adeel, 2025). P, D, and U, along with the credit assignment (reward) coming from the higher perceptual layer (teacher), can be seen as dynamic local competitive normalization and global cooperative organisation, respectively. This ensures that local and global coherence and consistency are maximized (Marvan and Phillips, 2024), while prediction error or free energy (Friston, 2005, 2010) is minimized, enabling a deeper form of "real understanding". A combination of three TPNs and one loop constitutes one agent. A set of 12 agents with 12 loops runs in parallel, evolving their Qs, Ks, and Vs simultaneously, before applying latent self-attention at $O(L \times N)$ where L is a small fraction of the input sequence length, making the overall cost approximately $O(N)$.

- Entity Tracking: Probes a model's capacity to update and maintain the state of entities throughout a narrative or dialogue by asking it to predict an entity's final condition after a series of changes (Kim and Schuster, 2023).
- EWoK: This benchmark evaluates the model's internal world knowledge across domains like spatial relations and social interactions (Ivanova et al., 2024).
- BLiMP: Testing various grammatical phenomenon, the Benchmark of Linguistic Minimal Pairs evaluates whether a model consistently picks the grammatically correct alternative from a pair of minimally different sentences (Warstadt et al., 2020).
- BLiMP Supplement: This is a supplement to BLiMP and was introduced in the first edition of the BabyLM challenge. It is more focused on dialogue and questions (Warstadt et al., 2025).

The metrics used to evaluate the model on each of these zero-shot benchmarks are as follows:

- Accuracy in predicting the correct completion or sentence for BLiMP, BLiMP Supplement, EWoK, Entity Tracking, and WUGs.
- Change in R^2 prediction from baseline for Eye Tracking and Self-paced Reading.

Table 1 shows the performance of tiny Co^4 language model on the metrics outlined above. As shown, our computationally efficient model, Co^4-α, outperforms GPT-2 on 5 out of 7 metrics. As for

[4]For the mathematical details of these functions and the core mechanism behind triadic modulation loops, please check (Graham et al., 2025).

GPT-BERT, another configuration Co^4-β, outperforms it on 4 out of 7 metrics. These hyperparameters for these configurations are further outlined in the Appendix.

Metric	GPT-2	Co^4-α	GPT-BERT	Co^4-β
Eye Tracking	8.66	**8.67**	**9.89**	8.19
Self-paced Reading	4.34	**4.59**	3.45	**3.62**
WUGs	52.50	**68.00**	43.00	**93.00**
Entity Tracking	13.90	**26.71**	33.96	**41.36**
EWoK	49.90	**50.01**	49.49	**50.11**
BLiMP	**66.36**	53.55	**71.66**	51.20
BLiMP Supplement	**57.07**	52.59	**63.21**	49.82

Table 1: **Zero-shot metrics comparison:** GPT-2 vs. Co^4-α and GPT-BERT (causal-focus) vs Co^4-β The single-layer, tiny Co^4 model outperformed GPT-2 on 5 out of 7 metrics, and GPT-BERT on 4 out of 7 metrics, despite being trained at a fraction of the computational cost, **in 2 epochs**.

Metric	GPT-2	GPT-BERT	Co^4-γ
Hypernym	48.93	49.05	**54.75**
QA Cong. Easy	50.00	67.19	**87.50**
QA Cong. Tricky	39.39	50.30	**53.94**
Subject Aux Inversion	**81.33**	81.28	65.48
Turn Taking	65.71	**68.21**	50.36
Overall	57.07	**63.21**	62.40

Table 2: **BLiMP Supplement benchmark**: Co^4-γ demonstrates superior performance in the BLiMP Supplement benchmark and the individual tasks in this benchmark. Although this configuration of Co^4-γ does not outperform the psycholinguistic metrics, it outperforms the baselines in the BLiMP Supplement.

Table 2 reports performance of Co^4-γ on the BLiMP Supplement benchmark. This Co^4-γ is a different configuration of our architecture, which notably performed better on BliMP Supplement. Since it did not beat most of the metrics, we did not pick it as our best configuration but we wanted to include its superior performance on BLiMP. It should be noted that our model performs better on BLiMP Supplement compared to BLiMP, suggesting that the Co^4 model has an inherent bias toward more complex tasks and long-term dependencies characteristic of BLiMP Supplement's subtasks. More challenging than the original BLiMP benchmark, BLiMP Supplement was introduced in

Task	Metric	GPT-2	GPT-BERT	Co^4
MRPC	F1	80.77	83.44	**84.15**
QQP	F1	62.45	**72.03**	62.73
BoolQ	Accuracy	66.91	68.07	**69.05**
MNLI	Accuracy	**51.12**	46.86	44.25
MultiRC	Accuracy	65.72	**68.28**	66.01
RTE	Accuracy	56.83	56.12	**59.71**
WSC	Accuracy	61.54	65.38	**67.31**

Table 3: SuperGLUE tasks

the most recent version of the BabyLM Challenge (Charpentier et al., 2025). It is more challenging since models perform relatively lower on it as compared to BLiMP (Warstadt et al., 2025), and also because it consists of more dialogues and questions as compared to the minimally different sentences in BLiMP. It is comprised of the following five subtasks:

- Hypernym: Checks whether a word is correctly recognized as a superset or subset of another (e.g., a dog is a mammal, so having a dog implies having a mammal).
- QA Congruence Easy: Verifies whether the question type matches the answer (e.g., a who question is answered with a person rather than a thing).
- QA Congruence Tricky: Similar to QA Congruence Easy but with more ambiguous cases.
- Subject–Aux Inversion: Checks whether the auxiliary verb is correctly inverted with the subject (e.g., Is she coming?).
- Turn Taking: Checks whether the correct personal pronoun is used when answering a question in dialogue.

Finetuning: Table 3 reports performance on SuperGLUE tasks as part of fine-tuning. (Wang et al., 2019). We picked our best Co^4 configuration overall (Co^4-α) for the finetuning. Our novel architecture achieves comparable results across most fine-tuning tasks and demonstrates better performance on 6 out of the 7 tasks when compared to GPT-2 and 4 out of the 7 tasks when compared to GPT-BERT. These tasks are:

- BoolQ: A yes/no question-answering dataset with unprompted and unconstrained questions (Clark et al., 2019)

- MNLI: The Multi-Genre Natural Language Inference corpus tests whether a model can recognize textual entailment (Williams et al., 2017).
- MRPC: The Microsoft Research Paraphrase Corpus contains pairs of sentences that are either paraphrases (semantically equivalent) or unrelated (Dolan and Brockett, 2005).
- QQP: Similarly to MRPC, the Quora Question Pairs corpus tests a model's ability to determine whether pairs of questions are semantically similar. These questions are sourced from Quora (BabyLM Community, 2023).
- MultiRC: The Multi-Sentence Reading Comprehension corpus evaluates a model's ability to select the correct answer from a list of candidates given a question and a context paragraph. In this version, the data is reformulated as a binary classification task judging whether an answer to a question-context pair is correct (Khashabi et al., 2018).
- RTE: Recognizing Textual Entailment tests the model's ability to recognize textual entailment (Dagan et al., 2005, 2022; Bentivogli et al., 2009).
- WSC: The Winograd Schema Challenge evaluates coreference resolution in sentences containing a pronoun and a list of noun phrases. This version reformulates the task as a binary classification problem using examples consisting of a pronoun and a noun phrase (Levesque et al., 2012).

The hyperparameters for this task are outlined in the Appendix.

4 Conclusion

The Co^4 model has a computational complexity of $O(L \cdot N + \alpha)$, scaling linearly with the number of input tokens (N), where L is the number of latent queries and α is a small fixed overhead. In contrast, models like GPT-2 and GPT-BERT scale quadratically at $O(N^2)$, making them significantly more expensive as input size grows. In standard Transformers, multiply–accumulate (MAC) operations grow with the quadratic term $P^2 \cdot E$ due to self-attention, where P is the number of tokens and E is the embedding dimension. In Co^4, this is replaced by a more efficient linear term $L_q \cdot P \cdot E$, enabled by a small set of latent queries. As a result, Co^4 achieves substantial computational savings and superior scalability over conventional Transformers. Despite being a single-layer model, the tiny Co^4 machine outperforms GPT-2 and GPT-BERT on most evaluated performance metrics, while requiring only a fraction of the computational resources. Future directions include scaling to larger datasets, integrating multi-objective or hybrid cost functions (e.g., those used in GPT-BERT), and evaluating different modes of apical operation (Phillips et al., 2024; Graham et al., 2024; Pastorelli et al., 2023). In addition, scaling beyond 8M parameters is part of ongoing work.

5 Acknowledgments

Advanced Research + Invention Agency (ARIA): Nature Computes Better Opportunity seeds. Professor Bill Phillips, Professor Leslie Smith, Professor Bruce Graham, and Dr Burcu Can Buglalilar from the University of Stirling. Professor Panayiota Poirazi from IMBB-FORTH, Professor Peter Konig from the University Osnabruck. Professor Heiko Neumann from Ulm University, Dr James Kay from the University of Glasgow, and several other eminent scholars for their help and support in several different ways, including reviewing this work, appreciation, and encouragement. We also acknowledge ChatGPT for its assistance with proofreading.

Competing interests The authors declare no conflict of interest.

References

Ahsan Adeel. 2020. Conscious multisensory integration: Introducing a universal contextual field in biological and deep artificial neural networks. *Frontiers in Computational Neuroscience*, 14.

Ahsan Adeel. 2025. Beyond attention: Toward machines with intrinsic higher mental states. *arXiv preprint arXiv:2505.06257*.

Ahsan Adeel, Adewale Adetomi, Khubaib Ahmed, Amir Hussain, Tughrul Arslan, and William A Phillips. 2023. Unlocking the potential of two-point cells for energy-efficient and resilient training of deep nets. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(3):818–828.

Ahsan Adeel, Mario Franco, Mohsin Raza, and Khubaib Ahmed. 2022. Context-sensitive neocortical neurons transform the effectiveness and efficiency of

neural information processing. *arXiv preprint arXiv:2207.07338*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Jaan Aru, Mototaka Suzuki, and Matthew Larkum. 2021. Cellular mechanisms of conscious processing. *Trends in Cognitive Sciences*, 25.

BabyLM Community. 2023. BabyLM Baseline 10M GPT-2. https://huggingface.co/BabyLM-community/babylm-baseline-10m-gpt2. Accessed: 2024-06-27.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1.

Jacopo Bono and Claudia Clopath. 2017. Modeling somatic and dendritic spike mediated plasticity at the single neuron and network level. *Nature communications*, 8(1):706.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, and 1 others. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *arXiv preprint arXiv:2502.10645*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Ido Dagan, Dan Roth, Fabio Zanzotto, and Mark Sammons. 2022. *Recognizing textual entailment: Models and applications*. Springer Nature.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.

Karl Friston. 2005. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836.

Karl Friston. 2010. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138.

Bruce P Graham, Jim W Kay, and William A Phillips. 2024. Transfer functions for burst firing probability in a model neocortical pyramidal cell. *bioRxiv*, pages 2024–01.

Bruce P Graham, Jim W Kay, and William A Phillips. 2025. Context-sensitive processing in a model neocortical pyramidal cell with two sites of input integration. *Neural Computation*, 37(4):588–634.

Alberto Granato, William A Phillips, Jan M Schulz, Mototaka Suzuki, and Matthew E Larkum. 2024. Dysfunctions of cellular context-sensitivity in neurodevelopmental learning disabilities. *Neuroscience & Biobehavioral Reviews*, page 105688.

et al. Greedy, Will. 2022. Single-phase deep learning in cortico-cortical networks. *Advances in Neural Information Processing Systems*.

Jordan Guerguiev, Timothy Lillicrap, and Blake Richards. 2017. Towards deep learning with segregated dendrites. *eLife*, 6:e22901.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19).

Michael Häusser. 2001. Synaptic function: Dendritic democracy. *Current Biology*, 11(1):R10–R12.

B Illing, J Ventura, G Bellec, and W Gerstner. 2022. Local plasticity rules can learn deep representations using self-supervised contrastive predictions. *Advances in Neural Information Processing Systems*.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, and 1 others. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

George Kastellakis, Alcino J Silva, and Panayiota Poirazi. 2016. Linking memories across time via neuronal and dendritic overlaps in model neurons with active dendrites. *Cell reports*, 17(6):1491–1504.

Jim W Kay and William A Phillips. 2020. Contextual modulation in mammalian neocortex is asymmetric. *Symmetry*, 12(5):815.

Jim W Kay, Jan M Schulz, and William A Phillips. 2022. A comparison of partial information decompositions using data from real and simulated layer 5b pyramidal cells. *Entropy*, 24(8):1021.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. *arXiv preprint arXiv:2305.02363*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Konrad P Körding and Peter König. 2000. Learning with two sites of synaptic integration. *Network: Computation in neural systems*, 11(1):25–39.

Matthew Larkum. 2013. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in neurosciences*, 36(3):141–151.

Matthew E Larkum. 2022. Are dendrites conceptually useful? *Neuroscience*, 489:4–14.

Matthew E Larkum, Lucy S Petro, Robert NS Sachdev, and Lars Muckli. 2018. A perspective on cortical layering and layer-spanning neuronal elements. *Frontiers in neuroanatomy*, 12:56.

Matthew E Larkum, J Julius Zhu, and Bert Sakmann. 1999. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–341.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. *KR*, 2012:13th.

Thomas Limbacher and Robert Legenstein. 2020. Emergence of stable synaptic clusters on dendrites through synaptic rewiring. *Frontiers in computational neuroscience*, 14:57.

Guy Major, Matthew E Larkum, and Jackie Schiller. 2013. Active properties of neocortical pyramidal neuron dendrites. *Annual review of neuroscience*, 36:1–24.

Tomáš Marvan and William A Phillips. 2024. Cellular mechanisms of cooperative context-sensitive predictive inference. *Current Research in Neurobiology*, page 100129.

Tomaš Marvan, Michal Polák, Talis Bachmann, and William A Phillips. 2021. Apical amplification—a cellular mechanism of conscious perception? *Neuroscience of consciousness*, 2021(2):niab036.

Andrew D Nelson and Kevin J Bender. 2021. Dendritic integration dysfunction in neurodevelopmental disorders. *Developmental Neuroscience*, 43(3-4):201–221.

Elena Pastorelli, Alper Yegenoglu, Nicole Kolodziej, Willem Wybo, Francesco Simula, Sandra Diaz, Johan Frederik Storm, and Pier Stanislao Paolucci. 2023. Two-compartment neuronal spiking model expressing brain-state specific apical-amplification,-isolation and-drive regimes. *arXiv preprint arXiv:2311.06074*.

Alexandre Payeur, Jordan Guerguiev, Friedemann Zenke, Blake A Richards, and Richard Naud. 2021. Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *Nature neuroscience*, 24(7):1010–1019.

W. A. Phillips, T. Bachmann, W. Spratling, L. Muckli, L Petro, and T. Zolnik. 2024. Cellular psychology: relating cognition to context-sensitive pyramidal cells. *Trends in Cognitive Sciences*.

William A Phillips. 2017. Cognitive functions of intracellular mechanisms for contextual amplification. *Brain and Cognition*, 112:39–53.

William A Phillips. 2023. *The Cooperative Neuron: Cellular Foundations of Mental Life*. Oxford University Press.

Panayiota Poirazi and Athanasia Papoutsi. 2020. Illuminating dendritic function with computational models. *Nature Reviews Neuroscience*, 21:1–19.

Srikanth Ramaswamy and Henry Markram. 2015. Anatomy and physiology of the thick-tufted layer 5 pyramidal neuron. *Frontiers in cellular neuroscience*, 9:233.

Rajesh PN Rao and Dana H Ballard. 1999. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87.

Mohsin Raza and Ahsan Adeel. 2024. An overlooked role of context-sensitive dendrites. *arXiv preprint arXiv:2408.11019*.

João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. 2018. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in neural information processing systems*, 31.

Jan M Schulz, Jim W Kay, Josef Bischofberger, and Matthew E Larkum. 2021. Gaba b receptor-mediated regulation of dendro-somatic synergy in layer 5 pyramidal neurons. *Frontiers in cellular neuroscience*, 15:718413.

Benjamin Schuman, Shlomo Dellal, Alvar Prönneke, Robert Machold, and Bernardo Rudy. 2021. Neocortical layer 1: An elegant solution to top-down and bottom-up integration. *Annual Review of Neuroscience*, 44(1):221–252. PMID: 33730511.

James M Shine. 2019. Neuromodulatory influences on integration and segregation in the brain. *Trends in cognitive sciences*, 23(7):572–583.

James M Shine, Patrick G Bissett, Peter T Bell, Oluwasanmi Koyejo, Joshua H Balsters, Krzysztof J Gorgolewski, Craig A Moodie, and Russell A Poldrack. 2016. The dynamics of functional brain networks: integrated network states during cognitive task performance. *Neuron*, 92(2):544–554.

James M Shine, Michael Breakspear, Peter T Bell, Kaylena A Ehgoetz Martens, Richard Shine, Oluwasanmi Koyejo, Olaf Sporns, and Russell A Poldrack. 2019. Human cognition involves the dynamic integration of neural activity and neuromodulatory systems. *Nature neuroscience*, 22(2):289–296.

James M Shine, Eli J Müller, Brandon Munn, Joana Cabral, Rosalyn J Moran, and Michael Breakspear. 2021. Computational models link cellular mechanisms of neuromodulation to large-scale neural dynamics. *Nature neuroscience*, 24(6):765–776.

Johan F Storm, P Christiaan Klink, Jaan Aru, Walter Senn, Rainer Goebel, Andrea Pigorini, Pietro Avanzini, Wim Vanduffel, Pieter R Roelfsema, Marcello Massimini, and 1 others. 2024. An integrative, multiscale view on neural theories of consciousness. *Neuron*, 112(10):1531–1552.

Mototaka Suzuki, Cyriel MA Pennartz, and Jaan Aru. 2023. How deep is the brain? the shallow brain hypothesis. *Nature Reviews Neuroscience*, 24(12):778–791.

Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. 2020. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, abs/2007.05558.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. *SuperGLUE: a stickier benchmark for general-purpose language understanding systems*. Curran Associates Inc., Red Hook, NY, USA.

Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. 2025. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjape, Adina Williams, Tal Linzen, and 1 others. 2025. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. *arXiv preprint arXiv:2504.08165*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR.

A Pre-Training Details

Hyperparameter	Co^4-α	Co^4-β	Co^4-γ
Number of parameters	8M	8M	8M
Number of layers†	1	1	1
Embedding size	256	256	256
Vocabulary size	16384	16384	16384
Attention heads	2	2	2
Hidden dropout	0.1	0.1	0.1
Batch size	32	64	32
Sequence length	512	512	512
Warmup ratio	1.3%	1.4%	1%
Learning rate	0.0002	0.00001	0.0002
Learning rate scheduler	constant	constant	cosine
Optimizer	ADAMW	ADAMW	ADAMW
ADAMW ϵ	1e-8	1e-8	1e-8
ADAMW β_1	0.9	0.9	0.9
ADAMW β_2	0.999	0.999	0.999

Table 4: Pre-training hyperparameters for the STRICT-SMALL track across three configurations. †One layer refers to a module composed of our custom Co4 layer.

The training procedure, which has been briefly highlighted before, is as follows. We use the same tokenizer as the baselines, with a vocab size of 16384 and a small 1-layer model with the hyperparameters mentioned above. The Co^4 language model with a single decoder layer and just two attention heads is trained on the 10M corpus. It is powered via the aforementioned triadic modulation loops among Q-, K-, and V-TPNs, operating through P, D, and U contextual fields. After token embedding and positional projection, each token's Q, K, and V vectors co-evolve through a series of rapid and modulated updates.

The main goal was to keep the model as minimal as possible, to see the true power of the biologically-inspired triadic modulation loops within the layer. It is observed that the model performance converges over just a few epochs, i.e., 2 in this case.

B Finetuning Details

We perform a grid search for the following hyperparameters:

- **Number of epochs:** {3, 5, 10}
- **Learning rate:** {3×10^{-5}, 5×10^{-5}, 1×10^{-4}, 2×10^{-4}, 3×10^{-4}, 5×10^{-5}, 5×10^{-5}}
- **Batch size:** {16, 32, 64}

For WSC (low training data), we expand the search to:

- **Number of epochs:** {3, 5, 10, 15, 20, 25, 30, 100}
- **Learning rate:** {3×10^{-5}, 5×10^{-5}, 7×10^{-5}, 1×10^{-4}, 2×10^{-4}, 3×10^{-4}, 5×10^{-4}}
- **Batch size:** {16, 32, 64} }

Teacher Demonstrations in a BabyLM's Zone of Proximal Development for Contingent Multi-Turn Interaction

Suchir Salhan* **Hongyi Gu** **Donya Rooein** **Diana Galvan-Sosa**
Gabrielle Gaudeau **Andrew Caines** **Zheng Yuan** **Paula Buttery**
ALTA Institute, Dept. of Computer Science & Technology, Cambridge University
NetMind.AI Bocconi University Sheffield University

Abstract

Multi-Turn dialogues between a child and caregiver are characterised by a property called CONTINGENCY – prompt, direct and meaningful exchanges between interlocuters. We introduce CONTINGENTCHAT, a Teacher–Student framework that benchmarks and improves multi-turn contingency in a BabyLM trained on 100M words. Using a novel alignment dataset for post-training, BabyLM generates responses that are more grammatical and cohesive. Experiments with adaptive teacher decoding strategies show limited additional gains. CONTINGENTCHAT highlights the positive benefits of targeted post-training on dialogue quality and indicates that CONTINGENCY remains a challenging goal for BabyLMs.

ContingentChat on HuggingFace (Models, Tokenizers and ContingentChat Post-Training Dataset)

Training, Post-Training & Analysis Code Open-Sourced on GitHub

1 Introduction

Conversational interaction with caregivers is crucial for children learning their first language (L1) or first languages (L1s). Linguistic interaction provides a source of primary linguistic data (PLD) for the learner, supporting the acquisition of formal competence of the target L1 grammar. It also serves as input for the acquisition of functional and pragmatic competence in the L1. A key feature of child-caregiver conversations to promote language learning is CONTINGENCY. Contingent interactions are the prompt and meaningful exchanges between a caregiver and infant that form the foundation for fluent and connected communication (Masek et al., 2021).

In this paper, we draw upon the notion of contingency in the context of cognitively-inspired small language modelling to design CONTINGENTCHAT, a cognitively-inspired post-training framework to enhance the contingency of BabyLM text generation in multi-turn dialogic interaction with a Teacher LLM. CONTINGENTCHAT is designed to enhance the dialogue generation capabilities of BabyLMs submitted for the STRICT track of the BabyLM Challenge, which are trained on 100M words of developmentally-plausible training data.

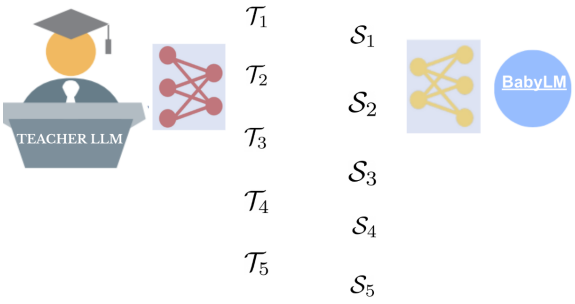


Figure 1: We consider multi-turn dialogic interactions between a BabyLM trained on 100M words (a STRICT model) and a Teacher LLM. The CONTINGENTCHAT framework aims to improve BabyLM generations by rewarding more cohesive and coherent generations through trials-and-demonstrations in a post-training phase.

Previous editions of the BabyLM Challenge (Warstadt et al., 2023; Hu et al., 2024) have evaluated submitted models trained with 10M (STRICT-SMALL) or 100M (STRICT) words on benchmarks of formal linguistic competence. These have included BLiMP (Warstadt et al., 2020), or BLiMP Supplement, which consist of minimal pairs designed to test the grammaticality judgements which language models are capable of making (Hu et al., 2024). However, as Charpentier and Samuel (2024) note, none of these benchmarks are well-suited for evaluating causal language model text generation, let alone evaluating generation quality or the alignment of model generations in an acquisition-inspired manner.

CONTINGENTCHAT is an iterative Teacher-Student post-training pipeline enabling interaction

*Corresponding Author: sas245@cam.ac.uk

Proceedings of the First BabyLM Workshop, pages 335–367

between a small language model (the *BabyLM*) and a much larger language model (the *Teacher*; see Figure 1). The BabyLM and Teacher LLM repeatedly interact on sub-dialogues selected from a 30M word annotated English dialogue corpus (**The CONTINGENTCHAT Alignment Dataset**). These 30M words come from the Switchboard Dialog Act Corpus (Godfrey et al., 1992; Stolcke et al., 2000), which we have annotated with cohesion metrics from different tools such as NLTK[1], Spacy[2], Tools for the Automatic Analysis of Cohesion 2.0 (TAACO) (Crossley et al., 2019), and our own bespoke processing and statistical calculations.

Our starting point in designing CONTINGENTCHAT is evaluating the dialogue generation capabilities of BabyLMs trained on 100M words of developmentally-plausible training data. We initially observed that STRICT BabyLMs are prone to self-repetition and struggle to produce coherent responses to prompts from a wide range of Teacher LLMs. We use our initial analysis to inform the experimental design of CONTINGENTCHAT.

In our framework, a BabyLM produces a continuation in a dialogue, and the Teacher LLM improves it according to strict anti-repetition and coherence guidelines. CONTINGENTCHAT rewrites BabyLM outputs to improve coherence and naturalness, which we treat as a chosen or edited response. This results in preference pairs – (1) original outputs of a BabyLM and (2) edited and improved responses by a Teacher LLM. CONTINGENTCHAT accumulates these preference pairs for post-training the BabyLM to gradually, over successive training rounds, guide it towards producing high quality, contextually-appropriate responses which are closer to the teacher's.

Our preliminary experiments indicate that models trained on fewer than 100M words struggle to sustain post-training interactions in multi-turn dialogues with a Teacher LLM. Despite achieving competitive scores on the BabyLM Evaluation benchmark, these smaller models often exhibit unstable or undesirable generation behaviours—such as self-repetition or incoherent responses—during post-training. Moreover, improvements from reward-based post-training appear inconsistent, suggesting that a minimum data scale is necessary for models to effectively benefit from reward learning and exhibit genuine intrinsic improvement.

[1] https://www.nltk.org/

[2] https://spacy.io/

Effective Teacher Demonstrations to a BabyLM are theoretised to be within a BabyLM's Zone of Proximal Development (ZPD). Vygotsky's Zone of Proximal Development (ZPD) proposes that learners are capable of acquiring new knowledge with support, up to the point at which such knowledge would be too complex to acquire (Vygotsky, 1978). Using CONTINGENTCHAT, we first systematically experiment in Experiment 1 with different post-training conditions that can potentially enhance the contingency of multi-turn BabyLM–LLM dialogues trained on 100M words. Experiment 2 assesses the benefits of using an adaptively-decoded Teacher LLM. This is motivated by one prevailing idea from language acquisition, known as the *Goldilocks Principle* (Kidd et al., 2014). Children naturally focus on input that is neither too simple nor too difficult but at the right level of challenge for learning, which suggests there might be benefits of adapting Teacher turns to the observed proficiency level of the BabyLM. We find scaffolding BabyLM outputs using reward models that encourage stricter adherence to Teacher Demonstrations forms a stable Zone of Proximal Development, effectively enhancing BabyLMs' capacity to generate contingent, contextually grounded dialogue via constrained policy updates.

2 Interactive Language Learning

2.1 Naturalistic Interaction in First Language (L1) Acquisition

In addition to acquiring formal competence of their first language, children have to learn to become competent conversational partners with others. This involves learning a complex set of skills. Spoken dialogues involve rapid exchanges of turns and interlocutors tend to use prediction and inference to keep a conversation flowing and coherent (Levinson, 2016).

Beyond learning specific skills like turn-taking in dialogues, contingency is a conversational behaviour that we define broadly, following Masek et al. (2021) and Agrawal et al. (2024b), as the ability to produce *multi-turn dialogues*[3]. Contingent di-

[3] Beyond the specific properties that characterise the quantity and quality of individuals language interactions, **contingency** can refer to the more general statistical learning process by which associations are formed between cues and outcomes (Ellis (2006a,b), Hsu et al. (2011) & Guo and Ellis (2021) i.a). This is a more general framing in the statistical learning literature which we apply in the more narrow domain of multi-turn dialogic interaction.

alogues have properties that distinguish them from successive chains of disconnected remarks or narrative monologues. Interlocutors have been theorised to operate via a Principle of Cooperativeness (Grice, 1975). Grice's Maxims of Conversation characterise some idealised characteristics of dialogue: where possible they should be informative, relevant, truthful, and clear. These maxims can be flouted or violated in adult speech for deliberate effect. More recent approaches in syntax and pragmatics have proposed varying theoretical analyses of the systematicity of dialogic interaction. Wiltschko (2021), for example, highlights that dialogic interaction is systematically driven by the dynamic and interactional process of finding a *mutual* common ground between interlocutors during a multi-turn dialogue. [4]

In the context of L1 acquisition, language acquisition researchers have suggested infants similarly demonstrate early systematic communicative behaviour, including via non-verbal cues like raised arms and deictic gestures (Heim and Wiltschko, 2025), as shown in example (1) from the Forrester Corpus (Forrester, 2002):

(1) a. *Ella: Whaaa↑ [raises arm] (1;00 – Forrester Corpus)*

b. *Ella: Yehh↑ [points to object with extended index finger]*

c. *Father: No, what's that? Huh?*

d. *Father: I don't know, do you?*

2.2 Interactive Language Learning

Work on deep multi-agent reinforcement learning (MARL) has demonstrated that agents can acquire complex behaviours – including emergent linguistic communication (Lazaridou et al., 2017) – by repeatedly interacting with other agents in a shared learning environment. These approaches leverage co-adaptation, allowing agents to bootstrap increasingly sophisticated behaviours without requiring complex simulators or demonstration data (Cao et al., 2018; Lu et al., 2020; Wang et al.; Lazaridou et al., 2020; Sadler et al., 2023).

Communication between artificial agents has been useful for investigating and simulating artificial language learning experiments that can be used to explore questions about learnability and the inductive biases of neural models (Lian et al., 2024, 2025; Kouwenhoven et al., 2024, 2025). Meanwhile, ter Hoeve et al. (2021) distinguishes Interactive Language Learning as interaction between a Teacher (a caregiver role) and Student LLM (whose role resembles a child) with interaction between them along with the environment that they share.

Our framework is inspired by the findings of Ma et al. (2024), whose trial-and-demonstration (TnD) learning framework showed that a student model benefits from the teacher's model choices. Their setting, however, targets word learning in language model training. CONTINGENTCHAT investigates the more complex task of generating individual responses which together build a coherent dialogue.

2.3 Zone of Proximal Development

One property of contingent multi-turn dialogues is that they are *mutual*. Caregivers are constantly adapting the contingency of outputs to keep conversations engaging during multi-turn dialogues (Hallart et al., 2022). Typically-developing L1 learners exhibit delays in reaching normal adult response times when engaging in multi-turn dialogues (Casillas, 2014). However, analysis of child-caregiver interactions shows that children aged between 1 and 3 years typically initiate simpler answers faster than more complex answers containing less familiar words (Casillas, 2014).

One possible realisation of contingent interaction from a caregiver might include adaptive lexical simplification to meet learner needs. This general behaviour resembles the Zone of Proximal Development (ZPD; Vygotsky (1978)), which in a general sense refers to the range of problems that a learner can solve with appropriate scaffolding but cannot tackle independently. Cui and Sachan (2025) apply this concept to design a curriculum for in-context learning with language models.

In our setting of Interactional Language Learning, we highlight that the ZPD is relevant in two distinct ways in contingent dialogues. Firstly, child-caregiver dialogues differ in substance between earlier and later learners, as the caregiver will estimate the ZPD of the learner. Secondly, within a multi-turn dialogue, contingency is a cycle of **anticipation** and **backtracking** as caregivers try to estimate and adaptively respond to the child's changing knowledge state and their ZPD. This re-

[4] Stalnaker (1978), Stalnaker (2002), Groenendijk and Roelofsen (2009) & Bavelas et al. (2012) i.a. also offer this interpretation of interactional language in terms of Common Ground (CG).

sults in strategies like conversational repair, including adult corrective moves which support L1 acquisition, and drawing mutual attention to form and meaning (Clark, 2020). Chouinard and Clark (2003) link adult reformulations directly to learning outcomes. The meta-pragmatic function of communicative feedback has been emphasised by Ben-Shlomo and Sela (2021) and Clark (2014) who highlight how the process of interpreting and responding to feedback can help children to learn about different types of feedback.

Other work categorises forms of repair (Norrick, 1991; Wilkinson and Weatherall, 2011; Cazden et al., 2017; Agrawal et al., 2024a,b, 2025). Feedback from caregivers to learners is neither random nor uniformly distributed across error types in L1 acquisition – early work by Hiller and Fernández (2016) find that certain error types, such as subject omission, attract more caregiver feedback, which has been supported by findings by Nikolaus et al. (2023) and Nikolaus (2023). Saxton (2000) and Saxton et al. (2005) highlight the sensitivity of child-to-adult corrections with immediate impacts of **direct negative evidence** in multi-turn dialogues and a delayed influence beyond immediate uptake.

SELF-REPAIR is an important mechanism of *metalinguistic awareness*, which Clark and Bernicot (2008) (i.a.) have argued scaffolds L1 acquisition by constructing a shared understanding in dialogic interaction, prompted by micro-cues from caregivers (e.g., `"hm?"`) that serve as subtle dialogic devices for self-correction. Forrester (2008) and Forrester and Cherington (2009) document the emergence of children's emerging self-repair capabilities as they internalise different feedback strategies.

3 Evaluating BabyLMs Multi-Turn Interaction

The aim of our proposed framework, CONTINGENTCHAT, is to characterise the qualities of contingent multi-turn dialogue between a Teacher LLM and a Student BabyLM trained on 100M words. We begin with a qualitative evaluation of the turns generated by student models submitted to the STRICT track of the BabyLM Challenge. Our motivation is to investigate whether the contingency of multi-turn dialogues can be improved by interaction.

We begin by analysing text generation by previous submissions to the BabyLM Challenge in two settings. First, we analyse the capabilities of models in generating continuations to an input prompt. Models appear to struggle to respond directly to prompts, as shown by a comparison of outputs from BabyLlama (Timiryasov and Tastet, 2023) and our baseline model that we pretrain. This is a 125M parameter Open Pre-Trained Transformer (OPT) (Zhang et al., 2022) trained on 100M words from the STRICT BabyLM Corpus (Charpentier et al., 2025).

> **Q:** What do you like to do in summer?
> **A: (BabyLlama)** What do you like to do in summer? Well, the winter season is about to begin. The months are longer than usual, and the temperatures are hotter. The months are hotter and colder. So, the months are longer and –
> **A (OPT-100M-base):** What's the matter?
> Nothing.
> That's all I'm doing.
> That's what I told you.
> Look, I'm getting so angry.
> If you're thinking about your mother...
> You're gonna get out of...

Impressionistically, these LLM outputs show poor contingency with the student model – failing to directly answer the prompt. We extend our preliminary analysis to investigate multi-turn dialogic interaction betweeen BabyLlama (Timiryasov and Tastet, 2023) and different Teacher LLMs (Gemini 2.5 Flash, Claude 3.7 Sonnet). Dialogues are presented in *Appendix* A and show similar behaviour to the examples above. In these preliminary experiments, we see improvements from introducing a **meta-prompt** which provides instructions to the Teacher Model – specifying its behaviour as a caregiver, the characteristics of the learner, the goals of the dialogues, and the ideal characteristics of multi-turn interaction.

3.1 Evaluating BabyLM Coherence and Contingency in Multi-Turn Dialogues

We go beyond impressionistic evaluation of the contingency of BabyLM outputs as a child/student in multi-turn Teacher-Student Interaction. Agrawal et al. (2024b) provide an automatic framework for evaluating contingency in CHILDES (MacWhinney and Snow, 1985). Their evaluation consists of three components: metrics that tag speech-act con-

gruence, semantic alignment between turns (measured using embeddings), and repetition (measured using SpaCy).

We propose that this evaluation strategy is potentially ill-suited to BabyLMs trained on 100M words, since our preliminary analysis shows that they are able to generate largely grammatical strings but struggle with the essential characteristics of contingency. Contingency, however, is an abstract concept that builds on more concrete linguistic elements, including lexical richness and cohesion. Considering the strong overlap between the notion of contingency and coherence, as defined in Linguistics[5], our framework relies on automatic metrics for the analysis of cohesion.

We introduce two contributions to evaluate contingency in multi-turn dialogues between a Teacher LLM and BabyLM. First, in *Section* 3.2, we develop our CONTINGENTCHAT Alignment Dataset based on discourse cohesion metrics. Secondly, we supplement this with human evaluation following Galvan-Sosa et al. (2025)'s Rubrik for evaluating LLM-generated text and explanations on the outputs of multi-turn Teacher-Student interaction.

3.2 The CONTINGENTCHAT Alignment Dataset

We capture text complexity differences through five complementary perspectives: semantic ambiguity, discourse connectives, syntactic complexity, cohesion, and lexical complexity. We draw on the `Switchboard Dialog Act Corpus` to compute different complexity metrics based on these five categories. For a sample, see Appendix B. We retrieve **lexical richness** (Vajjala and Meurers, 2012), type–token ratio (TTR), moving-average TTR (MATTR), and mean polysemy scores (mPOLY) as proxies for semantic ambiguity. **Discourse connectives** were quantified by the total number of connectives and the frequency of additive (e.g., "and", "also"), adversative (e.g., "but", "however"), and causal (e.g., "because", "therefore") subtypes (Pitler and Nenkova, 2009).

Syntactic complexity was measured by mean sentence length and mean clauses per sentence as indicators of structural elaboration (Chen and Zechner, 2011). **Cohesion** was assessed via lexical and grammatical overlap between adjacent sentences (content-word overlap and verb overlap) and by verb-tense repetition computed with TAACO- and NLTK-based taggers to capture temporal consistency. **Semantic and discourse features** included mean age of acquisition (how early words are typically learned), mean CEFR level (Common European Framework of Reference), concept density (distinct concepts per sentence), and an overall narrativity score indexing the extent to which a text exhibits narrative-like discourse (see Appendix D).

We use the 100M word Switchboard Corpus as a large-scale resource for metric estimation. This corpus contains transcribed English telephone conversations between speaker pairs in North America (Godfrey et al., 1992). We apply a turn segmentation procedure: utterances are first separated by speaker ID (A or B), consecutive utterances from the same speaker are merged into a single turn, and a "turn" is defined as any sequence of text transcribed from one speaker until there is a change of speaker. For dataset annotation, we sample exactly five turns per speaker, truncating the dialogue at that point and continuing segmentation across the whole corpus. Firstly, we compute the complexity metrics across these dialogues to compare metric distributions across speakers (see Figure 5).

3.3 Manual Evaluation

The main limitation of the metrics presented in *Section* 3.2 is that they were designed primarily for analyzing texts, narratives, and written discourse rather than conversational dialogue. To address this gap, we adapted the framework proposed by Galvan-Sosa et al. (2025) for explainability evaluation, which separates assessment into language and content dimensions.

Language features included GRAMMATICALITY (GRM), WORD CHOICE (WCH), and COHESION (COH), while content features encompassed CONCISENESS (CNC), APPROPRIATENESS (APP), and COHERENCE (COR). These feature definitions were adapted from an explainability context to the assessment of conversational contingency.

4 CONTINGENTCHAT Methodology

4.1 Rewarding Cohesive Response in Multi-Turn Interaction

Here we investigate two complementary training settings building on preference-based tuning. Experiment 1 uses a fixed teacher to generate improved continuations for student outputs, forming preference triples that fine-tune an OPT-style

[5] "The state of being logically consistent and connected" (Fetzer, 2012). It depends on a number of factors, including explicit cohesion cues.

causal LM (Zhang et al., 2022) with a Reference-Free Preference Objective. From each Switchboard dialogue in the CONTINGENTCHAT Alignment Dataset (*Section* 3.2), we extract one round (two turns) and append the next-speaker prefix to form a continuation prompt; the student samples a reply, and the teacher (`Llama-3.1-8B-Instruct`) produces a higher-quality alternative under instructions that discourage copying and enforce concise, coherent turns. We filter low-quality teacher outputs with automatic repetition checks, then optimize the student with an odds-ratio style preference objective in five disjoint iterations (dataset slices), carrying weights forward each round. In line with the idea of trial-and-demonstration tuition, this setting treats the teacher's alternative as a soft target that shapes the student's conversational form.

4.2 Adaptively-Decoded Teacher Demonstrations in a BabyLM's Zone of Proximal Development (ZPD)

While the benefits of including Child Directed Speech (CDS) in pretraining corpora are contested (Feng et al., 2024; Padovani et al., 2025), we attempt to lexically constrain the output from a Teacher LLM according to lexical curricula based on the Common European Framework of References for Language (CEFR) (Council of Europe, 2020). We hypothesise that this might simulate more contingent input from a caregiver and thus be a cognitively-inspired mechanism to create *opportune adaptive learning moments* for learners (Masek et al., 2021) in a teacher-student interaction. Experiment 2 replaces the teacher with a controllable ParlAI BlenderBot 3B model[6] and imposes a curriculum over linguistic complexity via CEFR levels. Motivated by the concept of a ZPD – learning is maximized when input difficulty is just beyond current independent performance – we constrain the teacher to generate at successive CEFR levels (A2→B1→B2→C1→C2) and, in a separate run, the reverse order, using the CEFR descriptors to operationalize difficulty level.

5 Experiments

5.1 Non-Interactive Baselines

5.1.1 Training Datasets

For our initial experiments we considered a range of pre-training datasets; here, we report experiments for OPT models trained on the STRICT BabyLM Corpus. In preliminary work, we pre-trained models on the **KidLM Corpus** (Nayeem and Rafiei, 2024) + **BabyLM Corpus**: The KidLM Corpus consists of 50.43M words of high-quality genre-diverse child-directed informational content, largely sourced from news articles. However, we found that models were unable to generate multi-turn dialogues in our post-training experiments. Pre-training corpora more aligned with those used in pre-trained dialogue agents might be more suitable for Teacher-Student Interaction (Zhang et al., 2020).

5.1.2 Architectures

We train a 125M OPT architecture with warm-up and a sequence length of 1024, which is found by Salhan et al. (2025) to be an optimal sequence length for pre-training BabyLMs. We also experiment with sequence lengths of 4096. See Appendix C for detailed experimental settings.

5.2 Experiment 1: Preference-Free Optimisation

Contrastive Preference Optimization (CPO) (Xu et al., 2024) and Monolithic Odds Ratio Preference Optimization (ORPO) (Hong et al., 2024) are two recent approaches for aligning language models with human preferences, but they differ fundamentally in methodology and applicability.

CPO extends Direct Preference Optimization (DPO) to train models to avoid producing translations that are adequate but suboptimal, addressing two key limitations of supervised fine-tuning (SFT): the performance ceiling imposed by reference-quality data and the lack of mechanisms to penalize disfavoured outputs. By leveraging contrastive comparisons between preferred and disfavoured outputs, CPO can be applied beyond machine translation to general domains such as dialogue. Under CPO, the student model would optimize to avoid generating replies that are adequate but suboptimal compared to the teacher's higher-quality alternative. This implies that the model might produce outputs that are conservatively aligned with the teacher, focusing on minimizing the contrastive loss derived from the teacher's preferred continuation. As a result, CPO is likely to yield responses that closely match the teacher's style and content, potentially at the cost of reduced diversity or creativity in dialogue. In contrast, ORPO eliminates the need for a separate reference model by directly

[6] https://huggingface.co/facebook/blenderbot-3B

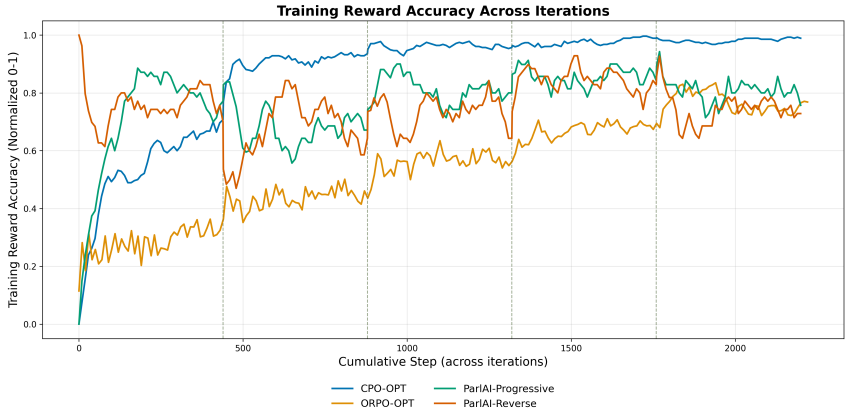


Figure 2: Reward accuracy during post-training of OPT (with 1024 sequence length) with CPO/ORPO (Experiment 1) and Progressive/Regressive CEFR (Experiment 2).

Model	BLIMP	BLIMP-S	COMPS	Entity	EWOK	Eye-Track	Self-Paced	WUG-Adj	WUG-Past	Average
cpo-opt-4096	55.59	48.08	50.27	**40.9**	49.82	0.26	**0.16**	0.61	0.0	27.299
cpo-opt-1024	75.74	67.73	55.51	26.69	51.36	0.33	0.03	0.66	0.04	30.899
orpo-opt-1024	75.04	66.2	**56.28**	26.98	51.42	0.2	0.02	0.68	0.06	30.764
orpo-opt-4096	55.71	47.73	50.13	40.75	49.33	0.2	0.1	0.65	-0.02	27.176
orpo-opt-100M-2048-preprocess	71.41	54.57	53.92	20.84	50.39	**1.07**	0.02	0.68	-0.02	28.098
cpo-opt-100M-2048-preprocess	71.43	54.65	53.95	20.83	50.3	**1.07**	0.02	0.68	-0.02	28.101
opt-base	70.45	55.18	54.28	24.1	50.83	0.45	0.03	**0.69**	**0.25**	28.473
opt-cefr-iteration1	**75.96**	67.73	55.9	26.78	**51.54**	0.2	0.05	0.67	0.02	**30.983**
opt-cefr-iteration2	75.6	67.84	55.54	26.92	51.19	0.19	0.06	0.66	0.03	30.892
opt-cefr-iteration3	75.48	67.36	55.4	27.08	51.33	0.19	0.06	0.66	0.02	30.842
opt-cefr-iteration4	75.27	67.59	55.39	26.79	51.49	0.19	0.06	0.65	0.03	30.829
opt-cefr-iteration5	75.18	67.22	55.44	26.54	51.23	0.19	0.06	0.65	0.03	30.727
opt-cefr-reverse-iteration4	75.32	67.7	55.43	26.51	51.5	0.19	0.06	0.65	0.02	30.82
opt-cefr-reverse-iteration5	75.25	67.08	55.45	26.41	51.43	0.19	0.06	0.65	0.03	30.728

Table 1: Evaluation results across different BabyLM Evaluation benchmarks (BLiMP, BLiMP Supplement, COMPS, Entity Tracking, EWoK, Eye Tracking and Self-Paced Reading Scores, WUG Adjective Nominalisation and Past Tense) for models in Experiments 1 and 2 compared to baselines. **Bolded** scores indicate highest accuracy (spearman rho correlation for Wug).

Model	AoA	CEFR	TTR	Rep.	Overlap	Norm. Avg
cpo-opt-4096	4.523	1.219	0.464	0.797	0.046	0.389
cpo-opt-1024	5.011	1.408	**0.624**	0.946	0.044	**0.496**
orpo-opt-1024	4.813	1.343	0.604	0.898	0.066	0.459
orpo-opt-4096	4.487	1.228	0.473	0.879	0.075	0.346
orpo-opt-100M-2048-preprocess	5.123	1.415	0.582	0.777	0.078	0.440
cpo-opt-100M-2048-preprocess	5.087	1.373	0.620	0.851	0.068	0.436
opt-base	**5.214**	**1.468**	0.590	0.881	**0.082**	0.425
opt-cefr-iteration1	4.813	1.325	0.627	0.803	0.044	0.386
opt-cefr-iteration2	4.802	1.371	0.610	0.870	0.053	0.454
opt-cefr-iteration3	4.782	1.305	0.606	0.904	0.082	0.469
opt-cefr-iteration4	4.951	1.328	0.599	0.887	0.062	0.465
opt-cefr-iteration5	4.819	1.328	0.572	0.909	0.056	0.465
opt-cefr-reverse-iteration4	4.948	1.331	0.620	**0.942**	0.062	0.458
opt-cefr-reverse-iteration5	4.929	1.322	0.596	0.867	0.063	0.399

Table 2: Evaluation Results on Cohesion Metrics of CONTINGENTCHAT Models from Experiment 1 and 2 (CEFR-based progressive and reverse iterations) against baseline checkpoints. Metrics: AoA (Age of Acquisition), CEFR (mean CEFR level), TTR (type–token ratio), Rep. (verb tense repetition), Overlap (content word overlap), and Norm. Avg (normalized average). Additional evaluation results can be found in *Appendix* E.

optimizing the odds ratio between favoured and disfavoured outputs within SFT. ORPO's monolithic formulation directly optimizes an odds-ratio preference objective without relying on a separate reference model, integrating the preference signal into the student's supervised fine-tuning process. Consequently, ORPO can more efficiently incorporate the teacher's preferred turn while allowing the student greater flexibility in phrasing, leading to responses that retain coherence and conciseness

while exploring alternative valid formulations. We hypothesise that ORPO may converge faster and produce a wider variety of acceptable replies across the five disjoint iterations, whereas CPO emphasizes stricter adherence to the teacher's guidance.

5.3 Experiment 2: Adaptively-Decoded Teacher Model

For our Teacher Model we follow Tyen et al. (2022) by adaptively decoding the difficulty of messages generated by a BlenderBot 3B model [7] according to the CEFR language proficiency framework [8]. This Controllable Complexity Teacher Model considers multiple candidate messages, before selecting the most appropriate one. We follow the default settings of Tyen et al. (2022) for re-ranking, except we use a smaller beam search size of 5. This generates 5 candidate messages from the Teacher Model for each turn. Tyen et al. (2022) train a regressor to predict the CEFR level of sentences. When the chatbot is in use, the regressor will predict the CEFR level of all candidate messages, allowing us to compute a score that combines the original ranking and the predicted CEFR. This score will then be used to re-rank the candidates, and the top candidate message will be sent to the user. Preference pairs (teacher "chosen" vs. student "rejected") are then used to update the student with a contrastive preference-optimization objective, allowing us to test whether training by complexity –via a CEFR-aware teacher – better aligns the student's dialogue behaviour with coherent, level-appropriate responses.

6 Evaluation

6.1 Task Evaluation and Post-Training Accuracy

We evaluate OPT models with sequence lengths of $\{1024, 4096\}$ on the BabyLM Evaluation Pipeline (Charpentier et al., 2025). Results are shown in Table 1. The first few rows show evaluation results for Experiment 2 with progressive CEFR alignnent (each iteration is where we progressively increase the CEFR level for adaptive decoding from the Teacher Model). The next few rows compare our OPT models with ORPO (Hong et al., 2024) and CPO (Xu et al., 2024). We also plot the accuracy of ORPO and CPO on the CONTINGENTCHAT Alignment dataset in *Figure* 2. Additional figures are found in the *Appendix* F.

[7] https://huggingface.co/facebook/blenderbot-3B

[8] https://github.com/WHGTyen/ControllableComplexityChatbot

Teacher	GRM	WCH	COH	CNC	APP	COR
Dialogue 1	✓	✗	✓	✗	✗	✗
Dialogue 2	✗	✗	✗	✗	✗	✗
Dialogue 3	✓	✗	✓	✗	✗	✗
Dialogue 4	✓	✗	✓	✗	✗	✗
Student	**GRM**	**WCH**	**COH**	**CNC**	**APP**	**COR**
cpo-opt-1024	✓	✗	✓	✗	✗	✗
cpo-opt-4096	✗	✗	✗	✗	✗	✗
orpo-opt-1024	✓	✗	✓	✗	✗	✗
orpo-opt-4096	✗	✗	✗	✗	✗	✗

Table 3: Qualitative judgements of Grammaticality, Word choice, Cohesion, Conciseness, Appropriateness and Coherence. Dialogues 1 - 5 refer to LLama 3.1B with corresponding student model.

6.2 Text Generation Evaluation

We evaluate our models using Cohesion Metrics and different meta-prompts that aim to simulate differences in dialogue generation characteristics.

Cohesion Metrics. Based on Table 2, cpo-opt-1024 achieves the highest normalized average score (0.496) among the CPO/ORPO variants, with strong lexical diversity (TTR = 0.624) and high repetition control (Rep. = 0.946), indicating robust overall performance. There are inconsistent benefits of CEFR-alignment. Worse performance might be due to limited beam search since Tyen et al. (2022) generate 20 responses per turn.

Human Evaluation. Following Galvan-Sosa et al. (2025)'s approach, each feature was manually assessed in a binary manner (yes/no) for each dialogue in the evaluation set generated using 5 conversation starters that were consistent with our preliminary dialogue generation.

While most of the dialogues were judged to be grammatical and cohesive, they failed to meet the rest of the features of contingency. Table 3 reports binary human judgements on 10 multi-turn dialogues with 8 turns generated between Llama-3.1B Teacher and four student models (`cpo-1024`, `cpo-4096`, `orpo-1024`, `orpo-4096`). This highlights the inherent complexity of conversational text, where lexical overlap within individual turns does not necessarily indicate that the dialogue as a whole achieved contingent interaction. Here, `cpo-opt-1024` achieves the best overall performance.

Meta-Prompts. Table 15 in the *Appendix* shows generation across interactions between teacher and student generated by `cpo-opt-1024` with meta-

prompts to the Teacher Model to generate dialogue starters based on specified age roles of the student model. Across most metrics, the 3–4 years group exhibits a more complicated linguistic profile in comparison to 2–3 years, with the highest Age of Acquisition (AoA) and CEFR level. Younger groups (6–11 months and 18–23 months) often occupy a middle ground, but show more overlap in aggregated data. With increasing interaction, we observe a drop in TTR and lexical richness (AoA/CEFR), and increases in cohesion and repetition (i.e., Overlap/Rep.).

7 Discussion

Our preliminary analysis of outputs from STRICT BabyLMs highlights a persistent gap between grammaticality and the communicative capabilities of BabyLMs, and we have presented contingency as one way to potentially improve interactional abilities. Our different experimental setups explore how **teacher demonstrations** can be utilised in multi-turn interaction. Experiment 1 uses an interactive setup that provides a measurable quantitative and qualitative improvement in turn-level coherence, lexical continuity, and grammatical repair across multi-turn Teacher-Student Interactions. It is possible to distinctly interpret ORPO and CPO post-training pipelines used in CONTINGENTCHAT in Vygotskian terms – both define and regulate a dynamic "scaffolded" learning region for the BabyLM (a Zone of Proximal Development) where communicative competence can be acquired through guided interaction.

The interplay of reward signals and policy constraints determines how far the student BabyLM may deviate from the behaviour of the Teacher LLM, while still being reinforced for progress toward more contingent, coherent, and human-like dialogue generation. The noticeable gains of the CPO reward model compared to ORPO are significant. CPO constrains the policy update to remain close to teacher demonstrations, effectively keeping the BabyLM's learning trajectory within a tightly scaffolded region of its ZPD. Throughout post-training, CPO anchors the updates of the BabyLMs more strongly than ORPO, potentially preventing premature drift into ungrounded or incoherent communicative behaviours. ORPO encourages exploration along preference gradients that are partially decoupled from the teacher's demonstrations, which could promote long-term generalisation and independence but also increases the likelihood of divergence from high-quality exemplars early on, leading to noisier learning dynamics or inconsistent contingent behaviour. In developmental terms, this potentially suggests that **BabyLMs might benefit from strong scaffolding via feedback that rewards improvement and maintains the student model's proximity to Teacher performance**.

In contrast, Experiment 2 revealed limited performance gains when the BabyLM is trained solely on static lexically-constrained Teacher Demonstrations without ongoing preference feedback. Although the model maintained grammatical competence and modestly improved surface-level coherence, it showed little advancement in deeper measures of contingency, such as pragmatic relevance and discourse-level alignment. This asymmetry suggests a crucial distinction between demonstrative and interactive scaffolding: while demonstrations expose the learner to appropriate communicative forms, they do not convey the adaptive feedback necessary to internalise when and why these forms should be used. Without the dynamic reinforcement provided in Experiment 1, the BabyLM might remain confined within its ZPD; capable of imitation, but unable to generalise beyond it. Further controlled experimentation is needed to confirm this hypothesis: for example, investigating different types of adaptive feedback that can improve contingency.

8 Conclusion

Our work demonstrates that contingency – prompt, direct, and meaningful exchanges – can be effectively benchmarked and improved in BabyLMs using the CONTINGENTCHAT Teacher–Student framework. Post-training with a carefully designed alignment dataset leads to more grammatical and cohesive multi-turn responses, while adaptive teacher decoding offers limited additional gains. The conditions for contingent dialogues from a BabyLM improve with **interactive scaffolding** and **adaptive feedback**, highlighting the benefits of continued ongoing, context-sensitive guidance that aligns learning signals with clear communicative goals. These results underscore the value of targeted post-training for enhancing dialogue quality and establish contingency as a meaningful and challenging objective for future BabyLM research.

Limitations

While CONTINGENTCHAT introduces a cognitively-motivated framework for enhancing contingency in small language models, several limitations constrain the generality and interpretability of our findings.

Post-Training Data and Domain. Our alignment dataset is derived exclusively from the Switchboard Dialog Act Corpus (Stolcke et al., 2000), which, although large and richly annotated, represents a narrow sociolinguistic domain—adult telephone conversations in American English. Consequently, the patterns of contingency learned during post-training may not generalise to other interactional contexts such as narrative discourse, spontaneous child-directed speech, or multilingual dialogue. Future work should extend our approach to corpora that more closely resemble early caregiver–child interactions or include non-Western varieties of English.

Limited Interpretability of Post-Training Reward-based fine-tuning may conflate linguistic and stylistic signals, making it challenging to disentangle which aspects of contingency are actually learned.

Experiments only with one Teacher Model All Student models were trained with feedback from a single Teacher LLM (Llama-3.1-8B-Instruct). This limits the robustness of our claims about the resulting contingent behaviour, as improvements may reflect stylistic imitation or alignment to that specific model's discourse patterns rather than generalized contingent competence. Investigation with more Teacher Models ecologically valid estimation of the Student's Zone of Proximal Development (ZPD).

Combination of Automatic and Human Evaluation Cohesion-based metrics (e.g., lexical overlap, verb repetition, CEFR-based lexical complexity) were originally developed for written text and do not fully capture pragmatic or conversational aspects of contingency such as repair, implicature, or turn-taking latency. Although we supplement these with human evaluation and dedicated significant effort to selecting automated metrics for evaluating dialogues, the resulting measures are imperfect proxies for the dynamic adaptivity that characterises natural dialogue.

Acknowledgements

The ALTA Institute authors are supported by Cambridge University Press & Assessment. Donya Rooein's research is supported through the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (No. 949944, INTEGRATOR). We would like to extend our thanks to Ethan Wilcox and Leshem Choshen, in particular, for their support during the review process, alongside the other BabyLM Workshop Organisers. This research was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council.

References

Abhishek Agrawal, Benoit Favre, and Abdellah Fourtassi. 2024a. Analysing communicative intent coordination in child-caregiver interactions. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Abhishek Agrawal, Benoît Favre, and Abdellah Fourtassi. 2025. Mapping the communicative landscape of early child-caregiver dialogue. *OSF*.

Abhishek Agrawal, Mitja Nikolaus, Benoit Favre, and Abdellah Fourtassi. 2024b. Automatic coding of contingency in child-caregiver conversations. In *LREC-COLING 2024*.

Janet Beavin Bavelas, Peter De Jong, Harry Korman, and Sarah Smock Jordan. 2012. Beyond back-channels: A three-step model of grounding in face-to-face dialogue. In *Proceedings of the Interdisciplinary Workshop on Feedback Behaviors in Dialogue*, pages 5–6, Stevenson, WA.

Ofira Rajwan Ben-Shlomo and Tal Sela. 2021. Conversational categories and metapragmatic awareness in typically developing children. *Journal of Pragmatics*, 172:46–62.

Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z Leibo, Karl Tuyls, and Stephen Clark. 2018. Emergent communication through negotiation. In *International Conference on Learning Representations*.

Marisa Casillas. 2014. Turn-taking. In *Pragmatic development in first language acquisition*, pages 53–70. John Benjamins.

Courtney B Cazden, Sarah Michaels, and Patton Tabors. 2017. Spontaneous repairs in sharing time narratives: The intersection of metalinguistic awareness, speech event, and narrative style. In *Communicative Competence, Classroom Interaction, and Educational Equity*, pages 99–113. Routledge.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Miao Chen and Klaus Zechner. 2011. Computing and evaluating syntactic complexity features for automated scoring of spontaneous non-native speech. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 722–731, Portland, Oregon, USA. Association for Computational Linguistics.

Michelle M Chouinard and Eve V Clark. 2003. Adult reformulations of child errors as negative evidence. *Journal of child language*, 30(3):637–669.

Eve V Clark. 2014. Two pragmatic principles in language use and acquisition. In *Pragmatic development in first language acquisition*, pages 105–120. John Benjamins Publishing Company.

Eve V Clark. 2020. Conversational repair and the acquisition of language. *Discourse Processes*, 57(5-6):441–459.

Eve V Clark and Josie Bernicot. 2008. Repetition as ratification: How parents and children place information in common ground. *Journal of child language*, 35(2):349–371.

Council of Europe. 2020. *Common European Framework of Reference for Languages: Learning, Teaching Assessment*, 3rd edition. StrasBourg.

Scott A Crossley, Kristopher Kyle, and Mihai Dascalu. 2019. The tool for the automatic analysis of cohesion 2.0: Integrating semantic similarity and text overlap. *Behavior research methods*, 51(1):14–27.

Peng Cui and Mrinmaya Sachan. 2025. Investigating the zone of proximal development of language models for in-context learning. *arXiv preprint arXiv:2502.06990*.

Nick C Ellis. 2006a. Language acquisition as rational contingency learning. *Applied linguistics*, 27(1):1–24.

Nick C Ellis. 2006b. Selective attention and transfer phenomena in l2 acquisition: Contingency, cue competition, salience, interference, overshadowing, blocking, and perceptual learning. *Applied linguistics*, 27(2):164–194.

Steven Y. Feng, Noah D. Goodman, and Michael C. Frank. 2024. Is child-directed speech effective training data for language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22055–22071, Miami, Florida, USA. Association for Computational Linguistics.

Anita Fetzer. 2012. Textual coherence as a pragmatic phenomenon.

Michael A. Forrester. 2002. Appropriating cultural conceptions of childhood: Participation in conversation. *Childhood*, 9(3):255–276.

Michael A Forrester. 2008. The emergence of self-repair: A case study of one child during the early preschool years. *Research on Language and Social Interaction*, 41(1):99–128.

Michael A Forrester and Sarah M Cherington. 2009. The development of other-related conversational skills: A case study of conversational repair during the early years. *First Language*, 29(2):166–191.

Diana Galvan-Sosa, Gabrielle Gaudeau, Pride Kavumba, Yunmeng Li, Hongyi Gu, Zheng Yuan, Keisuke Sakaguchi, and Paula Buttery. 2025. Rubrik's cube: Testing a new rubric for evaluating explanations on the CUBE dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23800–23839, Vienna, Austria. Association for Computational Linguistics.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society.

H. Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics, Volume 3: Speech Acts*, pages 41–58. Academic Press, New York.

Jeroen Groenendijk and Floris Roelofsen. 2009. Inquisitive semantics and pragmatics. In *Stanford workshop on Language, Communication and Rational Agency*, Stanford, CA. Paper presented May 30–31, 2009.

Rundi Guo and Nick C Ellis. 2021. Language usage and second language morphosyntax: Effects of availability, reliability, and formulaicity. *Frontiers in psychology*, 12:582259.

Charlie Hallart, Morgane Peirolo, Zihan Xu, and Abdellah Fourtassi. 2022. Contingency in child-caregiver naturalistic conversation: Evidence for mutual influence.

Johannes Heim and Martina Wiltschko. 2025. Rethinking structural growth: Insights from the acquisition of interactional language. *Glossa*.

Sarah Hiller and Raquel Fernández. 2016. A data-driven investigation of corrective feedback on subject omission errors in first language acquisition. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 105–114, Berlin, Germany. Association for Computational Linguistics.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Anne S Hsu, Nick Chater, and Paul MB Vitányi. 2011. The probabilistic analysis of language acquisition: Theoretical, computational, and experimental analysis. *Cognition*, 120(3):380–390.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Celeste Kidd, Steven T. Piantadosi, and Richard N. Aslin. 2014. The Goldilocks Effect in infant auditory attention. *Child Development*, 85:1795–1804.

Tom Kouwenhoven, Max Peeperkorn, Bram Van Dijk, and Tessa Verhoef. 2024. The curious case of representational alignment: Unravelling visio-linguistic tasks in emergent communication. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 57–71, Bangkok, Thailand. Association for Computational Linguistics.

Tom Kouwenhoven, Max Peeperkorn, and Tessa Verhoef. 2025. Searching for structure: Investigating emergent communication with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9977–9991, Abu Dhabi, UAE. Association for Computational Linguistics.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*.

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. 2020. Multi-agent communication meets natural language: Synergies between functional and structural language learning. *arXiv preprint arXiv:2005.07064*.

Stephen C Levinson. 2016. Turn-taking in human communication–origins and implications for language processing. *Trends in cognitive sciences*, 20(1):6–14.

Yuchen Lian, Arianna Bisazza, and Tessa Verhoef. 2025. Simulating the emergence of differential case marking with communicating neural-network agents.

Yuchen Lian, Tessa Verhoef, and Arianna Bisazza. 2024. NeLLCom-X: A comprehensive neural-agent framework to simulate language learning and group communication. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 243–258, Miami, FL, USA. Association for Computational Linguistics.

Yuchen Lu, Soumye Singhal, Florian Strub, Olivier Pietquin, and Aaron Courville. 2020. Supervised seeded iterated learning for interactive language learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3962–3970.

Ziqiao Ma, Zekun Wang, and Joyce Chai. 2024. Babysit a language model from scratch: Interactive language learning by trials and demonstrations. In *ICML 2024 Workshop on LLMs and Cognition*.

Brian MacWhinney and Catherine Snow. 1985. The child language data exchange system. *Journal of child language*, 12(2):271–295.

Lillian R Masek, Brianna TM McMillan, Sarah J Paterson, Catherine S Tamis-LeMonda, Roberta Michnick Golinkoff, and Kathy Hirsh-Pasek. 2021. Where language meets attention: How contingent interactions promote learning. *Developmental Review*, 60:100961.

Mir Tafseer Nayeem and Davood Rafiei. 2024. KidLM: Advancing language models for children – early insights and future directions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4813–4836, Miami, Florida, USA. Association for Computational Linguistics.

Mitja Nikolaus. 2023. *Communicative Feedback in Language Acquisition.* Ph.D. thesis, Aix-Marseille University.

Mitja Nikolaus, Laurent Prévot, and Abdellah Fourtassi. 2023. Communicative feedback in response to children's grammatical errors. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45.

Neal R Norrick. 1991. On the organization of corrective exchanges in conversation. *Journal of Pragmatics*, 16(1):59–83.

Francesca Padovani, Jaap Jumelet, Yevgen Matusevych, and Arianna Bisazza. 2025. Child-directed language does not consistently boost syntax learning in language models. *arXiv preprint arXiv:2505.23689*.

Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 13–16, Suntec, Singapore. Association for Computational Linguistics.

Philipp Sadler, Sherzod Hakimov, and David Schlangen. 2023. Yes, this way! learning to ground referring expressions into actions with intra-episodic feedback from supportive teachers. *arXiv preprint arXiv:2305.12880*.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2025. What is the Best Sequence Length for BabyLM? In *Proceedings of the BabyLM Workshop*, Suzhou, China. Association for Computational Linguistics.

Matthew Saxton. 2000. Negative evidence and negative feedback: Immediate effects on the grammaticality of child speech. *First Language*, 20(60):221–252.

Matthew Saxton, Phillip Backley, and Clare Gallaway. 2005. Negative input for grammatical errors: Effects after a lag of 12 weeks. *Journal of child language*, 32(3):643–672.

Robert Stalnaker. 1978. Assertion. In Peter Cole, editor, *Syntax and semantics, vol. 9: Pragmatics*, pages 315–332. Academic Press, Cambridge, MA.

Robert Stalnaker. 2002. Common ground. *Linguistics and Philosophy*, 25(5):701–721.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Maartje ter Hoeve, Evgeny Kharitonov, Dieuwke Hupkes, and Emmanuel Dupoux. 2021. Towards interactive language modeling. *arXiv preprint arXiv:2112.11911*.

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

Gladys Tyen, Mark Brenchley, Andrew Caines, and Paula Buttery. 2022. Towards an open-domain chatbot for language practice. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 234–249, Seattle, Washington. Association for Computational Linguistics.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the seventh workshop on building educational applications using NLP*, pages 163–173.

Lev Semenovich Vygotsky. 1978. *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Rundong Wang, Longtao Zheng, Wei Qiu, Bowei He, Bo An, Zinovi Rabinovich, Yujing Hu, Yingfeng Chen, Tangjie Lv, and Changjie Fan. Towards skill and population curriculum for marl.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Sue Wilkinson and Ann Weatherall. 2011. Insertion repair. *Research on Language and Social Interaction*, 44(1):65–91.

Martina Wiltschko. 2021. *The grammar of interactional language*. Cambridge University Press.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Shu Dewan, Marjan Ghazvininejad, Sinong Gutiérrez, Lucy Hazard, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

A Sample Dialogues

Gemini 2.5 Flash-BabyLLama

Q: What do you like to do in summer?
- It was a good day.
- This is a new day.
- What is it, a big party?
- It's just, like, a real big thing.
- What's this?
- It's really

Claude 3.7 Sonnet-BabyLLama

Q: What do you like to do in summer?
- It was a good day.
- This is a new day.
- What is it, a big party?
- It's just, like, a real big thing.
- What's this?
- It's really

Caregiver–Kid Communication Dialogue Meta-Prompt

Instructions: Assume the role of a caregiver. I will assume the role of a kid that's <INSERT AGE> years old. Your goal is to test the kid's communication skills through a dialogue, where you need to take the lead. Whenever it is your turn, consider that your responses should be no longer than one sentence.

You should aim to end the dialogue within 10 turns total (5 turns you, 5 turns me). When we reach the end, print a summary of our dialogue.

Start by asking: What do you like about summer?

B Sample CONTINGENTCHAT Alignment Dataset


```
{
  "dialogID": "dialog_02341.txt",
  "turns": [
    {
      "speaker": "B",
      "utterance": "And nothing is being done about it. Uh, the laws exist and
          are frequently upheld in, in, uh, in Appeals Court just because of
          technicalities and because of maybe small little holes that their
          defending attorney can find. And it's, it's really getting out of hand
           in many states."
    },
    {
      "speaker": "A",
      "utterance": "Well, the term technicality. The law enforcement community,
          uh, uh, you know, has to, has to separate the difference between
          somebody who is being set up in which, uh, grievous acts are done to,
          uh, to, you know, to get somebody into a, a situation where they're
          going to be guilty of, of a crime ..."
    },
    {
      "speaker": "B",
      "utterance": "Well, it seems like well it, it seems as if in the past
          typically there have been a lot of cases of people being wrongly tried
           or wrongly punished ...
    },
    {
      "speaker": "A",
      "utterance": "Uh-huh."
    },
    {
      "speaker": "B",
      "utterance": "And where his, old evidence was there, the witnesses were
          there, the, everything was conclusively pointing to this individual
          yet"
    }
  ],
}
```


Figure 3: Sample of the CONTINGENTCHAT ALIGNMENT Dataset

```
  "meta": {
    "length": 593,
    "ttr": {
      "noun": 0.162852,
      "verb": 0.154903,
      "adj": 0.182672
    },
    "type_token_ratios": [
      {
        "noun_ttr": 0.71,
        "verb_ttr": 0.475,
        "adj_ttr": 0.8571428571428571,
        "lemma_ttr": 0.332794830371567,
        "bigram_lemma_ttr": 0.8155339805825242,
        "trigram_lemma_ttr": 0.9708265802269044,
        "adjacent_overlap_all_sent": 0.1912442396313364,
        "lda_1_all_sent": 0.8396384935744969,
        "repeated_content_lemmas": 0.2116316639741518,
        "repeated_content_and_pronoun_lemmas": 0.2762520193861066
      },
      {
        "noun_ttr": 0.8166666666666667,
        "verb_ttr": 0.4561403508771929,
        "adj_ttr": 0.9444444444444444,
        "lemma_ttr": 0.3525179856115107,
        "bigram_lemma_ttr": 0.8269230769230769,
        "trigram_lemma_ttr": 0.9662650602409638,
        "adjacent_overlap_all_sent": 0.2067796610169491,
        "lda_1_all_sent": 0.8670341452328432,
        "repeated_content_lemmas": 0.1750599520383693,
        "repeated_content_and_pronoun_lemmas": 0.237410071942446
      },
      {
        "noun_ttr": 0.8857142857142857,
        "verb_ttr": 0.782608695652174,
        "adj_ttr": 1.0,
        "lemma_ttr": 0.5279187817258884,
        "bigram_lemma_ttr": 0.9183673469387756,
        "trigram_lemma_ttr": 0.9948717948717948,
        "adjacent_overlap_all_sent": 0.1742424242424242,
        "lda_1_all_sent": 0.8547770311665861,
        "repeated_content_lemmas": 0.116751269035533,
        "repeated_content_and_pronoun_lemmas": 0.182741116751269
      }
    ],
    "sentiment_scores": {
      "polarity": -0.473618,
      "subjectivity": -0.009093,
      "toxicity": 0.189254
    }
  }
}
```


Figure 4: Sample of the CONTINGENTCHAT ALIGNMENT Dataset

C Experimental Settings

C.1 Decoder Settings for Text Generation

Table 4: Decoding settings for Student and Teacher Generation.

Component	Parameter	Value
Student (child)	`max_new_tokens`	100
	`do_sample`	True
	`top_k`	50
	`top_p`	0.95
	`temperature`	0.8
	`num_return_sequences`	1
Teacher (LLM)	`max_new_tokens`	50
	`do_sample`	False

C.2 Training Hyperparameters shared across Experiments

Table 5: Preference optimization hyperparameters (ORPO and CPO; identical across experiments).

per-dev bsz	grad accum	eff. bsz	lr	epochs	warmup	max grad norm	fp16	grad ckpt
1	8	8	1×10^{-6}	1	10	0.5	False	True

optimizer	remove unused cols	drop last	num workers	save steps	eval steps	logging steps
`adamw_torch`	False	True	0	500	500	10

Table 6: Trainer setup for CPO and ORPO

C.3 ParlAI teacher (CEFR-controlled) configuration

Table 7: Key hyperparameters for ParlAI ControllableBlender teacher agent.

Parameter	Value / Description
Model Zoo	`blender_3B` (BlenderBot 3B)
Beam Size	20
Top-K Sampling	40
Rerank CEFR Level	dynamically set per ORPO phase (A2/B2/C1)
Rerank Tokenizer	`distilroberta-base`
Rerank Model	`complexity_model`
Rerank Model Device	`cuda`
Inference Mode	`rerank`
Filter Path	`data/filter.txt` (default)
Child Generation Args	max_new_tokens=50, do_sample=True, top_k=50 top_p=0.95, temperature=0.8
Teacher Generation Args	max_new_tokens=50, do_sample=False, temperature=0.3
Number of Prompts	8 (sampled per ORPO iteration)
Max Input Length	512 tokens (child fine-tuning)

C.4 Pretraining Hyperparameters

Parameter	Mamba	OPT
vocab_size	50280	50272
hidden_size	768	768
num_hidden_layers	32	12
state_size	16	–
expand / ffn_dim	2	3072
num_attention_heads	–	12
hidden_act	silu	relu

Table 8: Key default hyperparameters for MambaConfig and OPTConfig as implemented in Hugging Face Transformers.

D Linguistic Complexity Metrics

Metric	Abbrev.	Category	Description	Source
Type–Token Ratio	TTR	Lexical richness	Ratio of unique types to total tokens; indexes vocabulary diversity.	TAACO
Moving-Average TTR	MATTR	Lexical richness	Mean TTR over a sliding window to reduce text-length sensitivity.	TAACO
Mean polysemy	mPOLY	Lexical richness	Average meaningfulness scores for words in text	CRAT
Total discourse connectives	TDC	Discourse connectives	Count of connective tokens that explicitly link ideas across clauses/sentences.	manual
Additive connectives frequency	ACF	Discourse connectives	Rate of additive connectives (“and”, “also”, etc.).	manual
Adversative connectives frequency	AdCF	Discourse connectives	Rate of adversative connectives (“but”, “however”, etc.).	Spacy
Causal connectives frequency	CaCF	Discourse connectives	Rate of causal connectives (“because”, “therefore”, etc.).	manual
Mean sentence length	MSL	Syntactic complexity	Average number of tokens per sentence.	Spacy
Mean clauses per sentence	MCPS	Syntactic complexity	Average number of clauses per sentence.	Spacy
Content-word overlap (adjacent)	CWO-Adj	Cohesion	Proportion of content lemmas shared between adjacent sentences.	TAACO
Verb overlap (adjacent)	VO-Adj	Cohesion	Verb overlap between adjacent sentences	TAACO
Verb tense repetition (Repetition)	VTR	Cohesion	Share of adjacent sentences with matching verb tense (temporal consistency).	NLTK
Mean age of acquisition	AoA	Semantic	Average age at which words in the text are typically acquired.	CRAT
Mean CEFR level	CEFR	Semantic	Average CEFR level of words in text.	CRAT
Mean familiarity	MFam	Semantic	Average familiarity scores for words in text.	CRAT
Concept density	CD	Semantic	Number of concepts per sentence	spaCy
Narrativity score	Narr	Semantic	Composite narrativity score based on multiple metrics	manual

Table 9: Metrics used to assess linguistic complexity of the texts across five different categories.

Cohesion is computed as the average of all normalised TAACO metrics. Since all TAACO metrics range between 0–1, a simple mean provides an overall score; however, because some metrics consistently score near the top, examining their variance and distribution helps refine weighting. The figure shows the distributions of the 10 selected TAACO metrics.

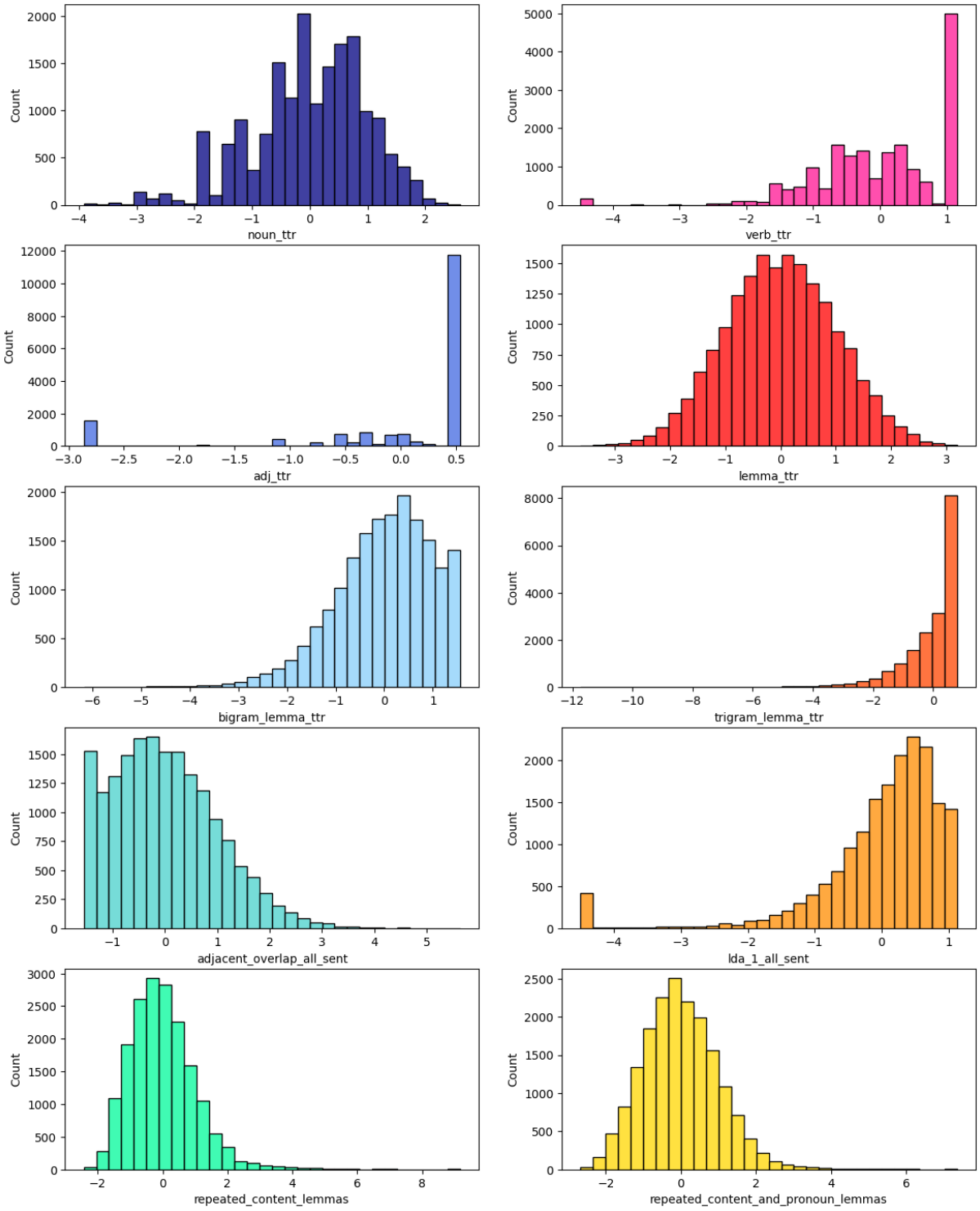


Figure 5: Distributions of the 10 selected TAACO metrics over the Switchboard dataset.

E Detailed Analysis of Teacher-Student Multi-Turn Dialogues

We include a detailed analysis of generated Teacher-Student Multi-Turn dialogues across different lengths. We include a more detailed breakdown of results summarised in *Table* 2. The teacher model, `meta-llama/Llama-3.2-3B-Instruct`, provides guidance and responses, while the student model, (e.g., `babylm-seqlen/opt-1024-warmup-v2`), is prompted and evaluated using the `facebook/opt-125m` tokenizer. We report results with $\{2, 4, 6, 8\}$ back-and-forth turns, with a maximum of $\{50, 100, 150, 200, 250\}$ tokens per turn.

Our generation scripts include utilities for cleaning generated responses, removing role tokens, unwanted punctuation, and other extraneous symbols, while also identifying banned tokens to avoid during generation. Teacher and student responses are generated using controlled sampling parameters such as top-p, top-k, temperature, and repetition penalties, with the student generation including multiple retry attempts to ensure meaningful output. The main generation function orchestrates multi-turn conversations, alternating between student and teacher turns, starting from a randomly selected conversation starter. The generated dialogues are structured with metadata, turn indices, and clean transcript text, and are finally saved as JSON files in a specified output directory. The script also includes a command-line interface allowing users to specify model IDs, tokenizers, number of turns, maximum token lengths, random seeds, devices, and output paths, making it versatile for experimentation and reproducible dialogue generation.

E.1 Conversation Starters

We provide the following conversation starters to generate dialogues between our Student and Teacher models. We apply automated metrics to these models.

```
{
  "STARTERS": [
    "Have you been on any trips recently? Where did you go, "
    "and did anything interesting happen there?",

    "What kind of music do you usually listen to? Do you have "
    "a favorite artist or concert experience you remember?",

    "Do you enjoy cooking at home? What's the best meal you've "
    "made recently, or do you prefer eating out?",

    "Do you have any pets? How long have you had them, and "
    "what do you like most about them?",

    "Do you play any sports or keep active? Have you joined any "
    "teams or tried something new lately?",

    "What's the weather usually like where you live? Does it affect "
    "your plans or the way you spend your weekends?",

    "Have you watched any shows or movies recently? Did you enjoy "
    "them, and would you recommend them to others?",

    "How's work going these days? Have you faced any interesting "
    "challenges or had any funny moments?",

    "Do you have any hobbies you like to spend time on? How did "
    "you get into them, and what keeps you interested?",

    "Do you celebrate any holidays with your family? Are there "
    "any special traditions or funny stories from past celebrations?"
  ]
}
```

Figure 6: Conversation starters used as initial prompts for multi-turn dialogue generation. Each starter is an open-ended question designed to elicit rich responses.

E.2 Extended Table 2 Results

Model	Turns	AoA	CEFR	Overlap	TTR	Rep.	NumCon	NormAvg
cpo_opt_100M_2048_preprocess	4	5.087	1.373	0.068	0.620	0.851	13.300	0.503
cpo_opt_base	4	5.214	1.468	0.074	0.590	0.881	13.100	0.556
cpo_opt_cosmos	4	5.067	1.383	0.052	0.638	0.912	14.200	0.530
cpo_opt_seqlen_1024_final_checkpoint	4	5.011	1.408	0.044	0.624	0.946	15.600	0.536
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	4	4.813	1.325	0.044	0.627	0.803	16.100	0.402
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	4	4.802	1.371	0.053	0.610	0.870	16.500	0.460
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	4	4.782	1.305	0.082	0.606	0.904	13.600	0.497
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	4	4.951	1.328	0.062	0.599	0.887	13.400	0.485
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	4	4.819	1.328	0.056	0.572	0.909	13.800	0.467
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	4	4.948	1.331	0.062	0.620	0.942	14.400	0.529
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	4	4.929	1.322	0.063	0.596	0.867	15.000	0.469
cpo_opt_seqlen_4096_final_checkpoint	4	4.523	1.219	0.046	0.464	0.797	18.100	0.286
mamba-sam-seqlen-2048-original	4	4.770	1.353	0.048	0.609	0.945	15.700	0.494
opt-sam-orpo-mamba-2048-step448	4	4.887	1.376	0.072	0.617	0.918	16.800	0.529
opt-sam-orpo-seqlen-2048-step559	4	4.948	1.365	0.052	0.615	0.866	15.100	0.474
opt-sam-seqlen-2048-original	4	4.843	1.318	0.052	0.618	0.942	17.000	0.503
orpo_opt_100M_2048_preprocess	4	5.123	1.415	0.078	0.582	0.777	14.300	0.467
orpo_opt_cosmos	4	5.200	1.459	0.067	0.624	0.822	10.300	0.514
orpo_opt_seqlen_1024_final_checkpoint	4	4.813	1.343	0.066	0.604	0.898	17.900	0.488
orpo_opt_seqlen_4096_final_checkpoint	4	4.487	1.228	0.075	0.473	0.879	11.500	0.371
babylm-seqlen-opt-1024-warmup-v2	4	4.864	1.343	0.057	0.594	0.925	16.500	0.496
babylm-seqlen-opt-4096-warmup-v2	4	4.487	1.172	0.030	0.449	0.867	17.100	0.291
cpo_opt_100M_2048_preprocess	6	5.071	1.423	0.065	0.564	0.830	19.000	0.479
cpo_opt_base	6	5.022	1.355	0.057	0.549	0.891	18.400	0.483
cpo_opt_cosmos	6	5.123	1.457	0.049	0.560	0.912	20.300	0.526
cpo_opt_seqlen_1024_final_checkpoint	6	4.947	1.405	0.050	0.563	0.911	25.900	0.498
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	6	4.898	1.350	0.050	0.554	0.915	23.400	0.478
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	6	4.812	1.341	0.067	0.543	0.922	25.200	0.488
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	6	4.812	1.337	0.083	0.531	0.893	20.300	0.481
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	6	4.882	1.339	0.057	0.536	0.938	22.700	0.491
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	6	4.886	1.333	0.064	0.528	0.929	22.600	0.490
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	6	4.846	1.331	0.036	0.557	0.916	21.300	0.450
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	6	4.982	1.373	0.069	0.541	0.853	23.000	0.471
cpo_opt_seqlen_4096_final_checkpoint	6	4.525	1.231	0.094	0.411	0.921	22.500	0.414
mamba-sam-seqlen-2048-original	6	4.881	1.325	0.059	0.558	0.969	29.200	0.522
opt-sam-orpo-mamba-2048-step448	6	4.831	1.369	0.084	0.553	0.912	24.200	0.514
opt-sam-orpo-seqlen-2048-step559	6	5.072	1.382	0.055	0.576	0.868	25.400	0.491
opt-sam-seqlen-2048-original	6	4.858	1.323	0.052	0.551	0.932	24.500	0.481
orpo_opt_100M_2048_preprocess	6	5.145	1.411	0.082	0.546	0.865	18.400	0.522
orpo_opt_cosmos	6	5.375	1.514	0.055	0.564	0.858	17.600	0.541
orpo_opt_seqlen_1024_final_checkpoint	6	4.951	1.390	0.065	0.567	0.927	26.100	0.526
orpo_opt_seqlen_4096_final_checkpoint	6	4.612	1.234	0.061	0.419	0.907	19.000	0.376
babylm-seqlen-opt-1024-warmup-v2	6	4.864	1.339	0.038	0.543	0.901	25.300	0.446
babylm-seqlen-opt-4096-warmup-v2	6	4.478	1.197	0.029	0.408	0.803	22.100	0.242
cpo_opt_seqlen_1024_final_checkpoint	8	4.917	1.346	0.058	0.523	0.893	40.100	0.476
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	8	4.786	1.307	0.063	0.507	0.894	27.500	0.444
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	8	5.018	1.400	0.063	0.524	0.881	30.600	0.491
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	8	4.790	1.324	0.066	0.489	0.842	30.500	0.413
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	8	4.858	1.346	0.075	0.510	0.885	34.000	0.475
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	8	4.874	1.345	0.070	0.500	0.948	29.700	0.506
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	8	4.889	1.355	0.070	0.519	0.856	29.700	0.456
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	8	4.868	1.337	0.059	0.508	0.850	30.200	0.428
cpo_opt_seqlen_4096_final_checkpoint	8	4.495	1.193	0.073	0.363	0.941	31.100	0.378
orpo_opt_seqlen_1024_final_checkpoint	8	4.958	1.359	0.076	0.511	0.930	40.000	0.526
orpo_opt_seqlen_4096_final_checkpoint	8	4.437	1.186	0.059	0.359	0.858	32.700	0.298
babylm-seqlen-opt-1024-warmup-v2	8	4.845	1.344	0.063	0.504	0.937	33.400	0.489
babylm-seqlen-opt-4096-warmup-v2	8	4.458	1.170	0.036	0.337	0.897	38.300	0.290

Table 10: Average metrics per BabyLM setting (Length = 50) with min–max normalized aggregate (NormAvg) across metrics.

Model	Turns	AoA	CEFR	Overlap	TTR	Rep.	NumCon	NormAvg
cpo_opt_100M_2048_preprocess	4	5.028	1.417	0.121	0.508	0.931	23.400	0.590
cpo_opt_base	4	5.075	1.404	0.088	0.496	0.847	21.000	0.491
cpo_opt_cosmos	4	5.196	1.458	0.075	0.540	0.876	25.100	0.540
cpo_opt_seqlen_1024_final_checkpoint	4	4.921	1.353	0.062	0.537	0.955	35.900	0.524
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	4	4.974	1.391	0.069	0.548	0.879	29.800	0.498
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	4	4.888	1.374	0.082	0.523	0.817	31.700	0.453
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	4	4.900	1.365	0.083	0.522	0.903	33.600	0.511
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	4	4.934	1.350	0.045	0.526	0.906	26.400	0.463
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	4	4.984	1.395	0.063	0.524	0.905	25.700	0.499
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	4	4.954	1.367	0.076	0.513	0.844	21.900	0.458
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	4	4.847	1.353	0.092	0.513	0.884	27.400	0.492
cpo_opt_seqlen_4096_final_checkpoint	4	4.532	1.224	0.058	0.391	0.891	33.300	0.349
mamba-sam-seqlen-2048-original	4	4.770	1.406	0.118	0.501	0.956	40.900	0.578
opt-sam-orpo-mamba-2048-step448	4	4.797	1.390	0.099	0.514	0.799	42.100	0.458
opt-sam-orpo-seqlen-2048-step559	4	4.728	1.304	0.077	0.512	0.893	40.100	0.463
opt-sam-seqlen-2048-original	4	4.917	1.439	0.089	0.523	0.904	35.500	0.538
orpo_opt_100M_2048_preprocess	4	5.021	1.419	0.085	0.524	0.862	20.300	0.503
orpo_opt_cosmos	4	5.211	1.443	0.091	0.532	0.866	26.100	0.549
orpo_opt_seqlen_1024_final_checkpoint	4	4.953	1.365	0.068	0.534	0.909	35.200	0.507
orpo_opt_seqlen_4096_final_checkpoint	4	4.711	1.262	0.062	0.419	0.870	19.600	0.371
babylm-seqlen-opt-1024-warmup-v2	4	4.878	1.342	0.110	0.493	0.847	38.800	0.492
babylm-seqlen-opt-4096-warmup-v2	4	4.378	1.189	0.067	0.341	0.770	35.000	0.238
cpo_opt_100M_2048_preprocess	6	5.093	1.437	0.090	0.483	0.858	36.100	0.517
cpo_opt_base	6	5.035	1.429	0.080	0.489	0.892	38.500	0.521
cpo_opt_cosmos	6	5.075	1.395	0.116	0.465	0.904	43.300	0.566
cpo_opt_seqlen_1024_final_checkpoint	6	4.904	1.364	0.094	0.456	0.932	55.000	0.535
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	6	4.843	1.325	0.099	0.453	0.901	51.400	0.501
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	6	4.805	1.323	0.092	0.460	0.892	49.000	0.482
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	6	4.916	1.107	0.112	0.465	0.876	42.900	0.461
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	6	4.943	1.127	0.078	0.486	0.888	40.100	0.439
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	6	4.845	1.366	0.095	0.469	0.912	45.600	0.514
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	6	4.953	1.363	0.068	0.469	0.915	44.500	0.495
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	6	4.974	1.394	0.075	0.483	0.858	42.800	0.478
cpo_opt_seqlen_4096_final_checkpoint	6	4.430	1.320	0.129	0.299	0.764	55.700	0.346
mamba-sam-seqlen-2048-original	6	4.838	1.429	0.105	0.474	0.840	59.100	0.503
opt-sam-orpo-mamba-2048-step448	6	4.959	1.411	0.066	0.478	0.962	60.700	0.549
opt-sam-orpo-seqlen-2048-step559	6	4.912	1.352	0.080	0.508	0.944	44.600	0.535
opt-sam-seqlen-2048-original	6	4.861	1.384	0.071	0.465	0.838	52.600	0.446
orpo_opt_100M_2048_preprocess	6	4.984	1.399	0.093	0.479	0.853	41.800	0.498
orpo_opt_cosmos	6	5.155	1.512	0.111	0.509	0.917	38.800	0.616
orpo_opt_seqlen_1024_final_checkpoint	6	4.965	1.382	0.069	0.467	0.888	57.200	0.492
orpo_opt_seqlen_4096_final_checkpoint	6	4.626	1.274	0.060	0.362	0.937	34.500	0.396
babylm-seqlen-opt-1024-warmup-v2	6	4.895	1.365	0.080	0.467	0.955	55.600	0.536
babylm-seqlen-opt-4096-warmup-v2	6	4.709	1.256	0.049	0.317	0.842	48.900	0.320
cpo_opt_seqlen_1024_final_checkpoint	8	4.956	1.406	0.113	0.421	0.953	69.300	0.584
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	8	4.938	1.389	0.081	0.436	0.863	69.200	0.485
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	8	4.954	1.402	0.083	0.435	0.898	63.500	0.512
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	8	5.062	1.417	0.090	0.440	0.917	63.100	0.551
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	8	4.939	1.116	0.102	0.437	0.893	60.400	0.466
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	8	5.110	1.419	0.084	0.431	0.930	55.600	0.550
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	8	5.052	1.159	0.085	0.448	0.940	56.200	0.500
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	8	4.903	1.104	0.108	0.431	0.879	58.300	0.453
cpo_opt_seqlen_4096_final_checkpoint	8	4.493	1.187	0.073	0.273	0.832	89.500	0.313
orpo_opt_seqlen_1024_final_checkpoint	8	5.052	1.424	0.072	0.443	0.957	74.700	0.564
orpo_opt_seqlen_4096_final_checkpoint	8	4.482	1.315	0.120	0.294	0.903	51.900	0.426
babylm-seqlen-opt-1024-warmup-v2	8	5.002	1.432	0.084	0.447	0.829	59.200	0.482
babylm-seqlen-opt-4096-warmup-v2	8	4.485	1.215	0.066	0.279	0.914	70.000	0.352

Table 11: Average metrics per BabyLM setting (Length = 100) with min–max normalized aggregate (NormAvg) across metrics.

Model	Turns	AoA	CEFR	Overlap	TTR	Rep.	NumCon	NormAvg
cpo_opt_100M_2048_preprocess	4	5.057	1.441	0.115	0.469	0.828	33.400	0.518
cpo_opt_base	4	5.153	1.433	0.082	0.458	0.883	34.500	0.519
cpo_opt_cosmos	4	5.174	1.460	0.076	0.488	0.894	33.800	0.539
cpo_opt_seqlen_1024_final_checkpoint	4	4.912	1.360	0.104	0.445	0.959	52.600	0.560
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	4	4.893	1.360	0.092	0.469	0.896	43.200	0.502
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	4	4.923	1.418	0.105	0.467	0.936	45.100	0.562
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	4	4.949	1.353	0.109	0.439	0.901	40.200	0.519
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	4	5.055	1.350	0.083	0.448	0.889	33.300	0.490
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	4	4.989	1.379	0.114	0.443	0.870	34.600	0.514
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	4	4.983	1.375	0.081	0.499	0.894	35.000	0.507
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	4	4.901	1.380	0.070	0.470	0.862	40.900	0.457
cpo_opt_seqlen_4096_final_checkpoint	4	4.385	1.256	0.128	0.302	0.749	49.200	0.311
mamba-sam-seqlen-2048-original	4	4.918	1.435	0.075	0.470	0.906	55.800	0.516
opt-sam-orpo-mamba-2048-step448	4	4.880	1.384	0.073	0.470	0.870	64.800	0.481
opt-sam-orpo-seqlen-2048-step559	4	4.910	1.371	0.069	0.473	0.921	47.300	0.500
opt-sam-seqlen-2048-original	4	4.998	1.416	0.117	0.466	0.913	45.500	0.570
orpo_opt_100M_2048_preprocess	4	5.169	1.432	0.109	0.473	0.766	33.300	0.483
orpo_opt_cosmos	4	5.284	1.489	0.102	0.507	0.847	27.000	0.561
orpo_opt_seqlen_1024_final_checkpoint	4	5.077	1.395	0.076	0.474	0.825	69.600	0.487
orpo_opt_seqlen_4096_final_checkpoint	4	4.732	1.301	0.078	0.391	0.855	37.700	0.395
babylm-seqlen-opt-1024-warmup-v2	4	4.778	1.351	0.098	0.441	0.903	55.800	0.497
babylm-seqlen-opt-4096-warmup-v2	4	4.473	1.222	0.032	0.355	0.900	50.300	0.316
cpo_opt_seqlen_1024_final_checkpoint	6	5.236	1.493	0.075	0.460	0.954	75.800	0.610
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	6	4.859	1.371	0.090	0.414	0.912	76.200	0.513
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	6	5.048	1.427	0.088	0.438	0.918	60.700	0.547
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	6	5.091	0.960	0.086	0.420	0.897	62.100	0.428
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	6	5.128	1.419	0.088	0.420	0.870	71.100	0.525
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	6	5.011	1.397	0.092	0.429	0.902	67.300	0.532
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	6	5.152	1.473	0.093	0.450	0.888	66.200	0.564
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	6	4.943	1.375	0.165	0.390	0.908	64.800	0.597
cpo_opt_seqlen_4096_final_checkpoint	6	4.410	1.191	0.070	0.252	0.885	78.400	0.320
orpo_opt_seqlen_1024_final_checkpoint	6	5.056	1.165	0.086	0.437	0.827	75.800	0.439
orpo_opt_seqlen_4096_final_checkpoint	6	4.621	1.283	0.077	0.306	0.876	58.000	0.374
babylm-seqlen-opt-1024-warmup-v2	6	5.060	1.429	0.076	0.439	0.858	66.000	0.499
babylm-seqlen-opt-4096-warmup-v2	6	4.537	1.213	0.063	0.266	0.890	89.300	0.347
cpo_opt_seqlen_4096_final_checkpoint	8	4.408	1.170	0.068	0.201	0.900	115.300	0.330
orpo_opt_seqlen_4096_final_checkpoint	8	4.481	1.202	0.070	0.248	0.890	85.200	0.338
babylm-seqlen-opt-4096-warmup-v2	8	4.505	1.218	0.036	0.251	0.856	100.000	0.293

Table 12: Average metrics per BabyLM setting (Length = 150) with min–max normalized aggregate (NormAvg) across metrics.

Model	Turns	AoA	CEFR	Overlap	TTR	Rep.	NumCon	NormAvg
cpo_opt_100M_2048_preprocess	4	5.171	0.932	0.108	0.432	0.904	40.300	0.453
cpo_opt_base	4	5.378	1.506	0.099	0.433	0.882	41.500	0.580
cpo_opt_cosmos	4	5.191	1.270	0.130	0.435	0.933	50.800	0.585
cpo_opt_seqlen_1024_final_checkpoint	4	4.960	1.394	0.105	0.427	0.891	67.500	0.533
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	4	4.827	1.345	0.113	0.430	0.946	59.900	0.547
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	4	4.952	1.391	0.061	0.419	0.874	58.100	0.457
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	4	5.184	1.474	0.115	0.434	0.873	60.400	0.576
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	4	4.899	1.337	0.071	0.469	0.872	44.300	0.458
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	4	5.025	1.387	0.093	0.441	0.904	50.500	0.527
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	4	5.006	1.444	0.068	0.454	0.887	43.300	0.494
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	4	5.163	1.257	0.083	0.453	0.875	50.100	0.488
cpo_opt_seqlen_4096_final_checkpoint	4	4.367	1.188	0.082	0.245	0.770	102.700	0.269
mamba-sam-seqlen-2048-original	4	4.967	1.547	0.108	0.430	0.819	73.800	0.529
opt-sam-orpo-mamba-2048-step448	4	5.049	1.435	0.095	0.438	0.853	76.100	0.525
opt-sam-orpo-seqlen-2048-step559	4	4.825	1.152	0.090	0.456	0.829	62.500	0.410
opt-sam-seqlen-2048-original	4	4.947	1.384	0.073	0.456	0.841	59.800	0.462
orpo_opt_100M_2048_preprocess	4	5.080	1.198	0.100	0.463	0.928	39.000	0.516
orpo_opt_cosmos	4	5.417	1.546	0.074	0.478	0.854	43.100	0.562
orpo_opt_seqlen_1024_final_checkpoint	4	5.047	0.888	0.094	0.438	0.948	71.000	0.463
orpo_opt_seqlen_4096_final_checkpoint	4	4.533	1.340	0.122	0.306	0.808	65.100	0.391
babylm-seqlen-opt-1024-warmup-v2	4	5.010	1.410	0.072	0.436	0.952	75.800	0.550
babylm-seqlen-opt-4096-warmup-v2	4	4.418	1.164	0.042	0.268	0.750	70.400	0.193
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	6	5.023	1.419	0.089	0.412	0.898	91.200	0.542
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	6	4.785	1.297	0.122	0.363	0.776	117.700	0.449
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	6	4.934	1.398	0.120	0.369	0.832	88.800	0.505
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	6	4.947	1.363	0.142	0.378	0.888	79.700	0.560
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	6	4.964	1.372	0.090	0.411	0.889	71.100	0.506
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	6	5.100	1.431	0.088	0.382	0.915	77.400	0.544
cpo_opt_seqlen_4096_final_checkpoint	6	4.510	1.267	0.084	0.244	0.809	98.300	0.328
orpo_opt_seqlen_4096_final_checkpoint	6	4.456	1.293	0.120	0.245	0.909	118.600	0.451
babylm-seqlen-opt-1024-warmup-v2	6	4.897	1.397	0.110	0.376	0.946	100.700	0.573
babylm-seqlen-opt-4096-warmup-v2	6	4.402	1.181	0.047	0.227	0.870	113.300	0.295
cpo_opt_seqlen_4096_final_checkpoint	8	4.366	1.223	0.109	0.196	0.934	198.200	0.464
orpo_opt_seqlen_4096_final_checkpoint	8	4.663	1.284	0.107	0.247	0.830	113.100	0.403
babylm-seqlen-opt-4096-warmup-v2	8	4.446	1.166	0.114	0.183	0.841	154.400	0.372

Table 13: Average metrics per BabyLM setting (Length = 200) with min–max normalized aggregate (NormAvg) across metrics.

Model	Turns	AoA	CEFR	Overlap	TTR	Rep.	NumCon	NormAvg
cpo_opt_100M_2048_preprocess	4	5.054	1.393	0.125	0.396	0.950	57.700	0.590
cpo_opt_base	4	5.655	1.647	0.108	0.411	0.835	47.300	0.622
cpo_opt_cosmos	4	5.124	1.435	0.104	0.419	0.855	61.500	0.531
cpo_opt_seqlen_1024_final_checkpoint	4	4.932	1.391	0.103	0.411	0.819	80.200	0.482
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration1	4	5.149	1.467	0.104	0.399	0.910	60.400	0.569
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration2	4	4.927	1.378	0.099	0.396	0.840	55.800	0.465
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration3	4	4.898	1.375	0.140	0.409	0.805	60.800	0.497
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration4	4	5.029	1.427	0.092	0.387	0.887	66.900	0.516
cpo_opt_seqlen_1024_progressive_cefr_parlai_iteration5	4	5.183	1.434	0.095	0.419	0.838	59.000	0.514
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration4	4	5.007	1.391	0.101	0.409	0.839	48.600	0.480
cpo_opt_seqlen_1024_progressive_cefr_reverse_parlai_iteration5	4	5.016	0.910	0.155	0.370	0.911	68.100	0.489
cpo_opt_seqlen_4096_final_checkpoint	4	4.632	1.245	0.094	0.286	0.891	78.300	0.405
mamba-sam-seqlen-2048-original	4	4.836	1.413	0.100	0.411	0.869	100.300	0.519
opt-sam-orpo-mamba-2048-step448	4	4.777	1.140	0.126	0.388	0.778	108.300	0.421
opt-sam-orpo-seqlen-2048-step559	4	5.091	1.454	0.092	0.407	0.861	67.500	0.520
opt-sam-seqlen-2048-original	4	4.997	1.439	0.116	0.401	0.907	71.200	0.565
orpo_opt_100M_2048_preprocess	4	5.160	1.247	0.101	0.424	0.897	47.700	0.510
orpo_opt_cosmos	4	5.386	1.497	0.120	0.436	0.844	47.600	0.585
orpo_opt_seqlen_1024_final_checkpoint	4	5.073	1.197	0.096	0.412	0.888	90.700	0.502
orpo_opt_seqlen_4096_final_checkpoint	4	4.741	1.391	0.117	0.318	0.810	49.500	0.416
babylm-seqlen-opt-1024-warmup-v2	4	4.977	1.426	0.077	0.429	0.905	71.000	0.520
babylm-seqlen-opt-4096-warmup-v2	4	4.549	1.223	0.070	0.266	0.922	79.400	0.374
cpo_opt_seqlen_4096_final_checkpoint	6	4.375	1.179	0.104	0.212	0.857	132.200	0.360
orpo_opt_seqlen_4096_final_checkpoint	6	4.498	1.226	0.151	0.214	0.932	130.800	0.492
babylm-seqlen-opt-4096-warmup-v2	6	4.298	1.118	0.070	0.168	0.780	182.400	0.264
cpo_opt_seqlen_4096_final_checkpoint	8	4.366	1.207	0.129	0.159	0.838	245.300	0.442
orpo_opt_seqlen_4096_final_checkpoint	8	4.434	1.283	0.141	0.181	0.938	254.000	0.561
babylm-seqlen-opt-4096-warmup-v2	8	4.531	1.193	0.063	0.171	0.715	162.200	0.245

Table 14: Average metrics per BabyLM setting (Length = 250) with min–max normalized aggregate (NormAvg) across metrics.

E.3 Effect of Meta-Prompts

We additionally report the effect of providing meta-prompts (using the template in *Section* A) to the Teacher Model on our automatic metrics. We report results for our best performing student model `cpo-opt-1024`.

Age	TurnNo.	AvgSentLen	AoA	AddCon	CEFR	Overlap	CausalCon	TTR	ConceptDensity
6-11m	1	21.304	5.080	3.600	1.390	0.201	0.700	0.483	5.399
	3	25.919	5.008	7.350	1.381	0.250	1.700	0.232	5.613
	5	29.969	4.940	10.500	1.381	0.286	3.050	0.163	5.666
	all	34.947	4.860	15.500	1.354	0.324	4.050	0.107	5.886
18-23m	1	15.193	4.734	2.750	1.296	0.307	0.900	0.454	4.213
	3	17.580	4.689	6.600	1.277	0.330	2.300	0.234	4.450
	5	17.818	4.689	10.900	1.272	0.352	3.750	0.164	4.422
	all	20.676	4.609	16.050	1.241	0.411	5.150	0.099	4.332
2-3y	1	17.008	4.534	3.550	1.209	0.112	0.400	0.554	4.039
	3	18.100	4.460	7.600	1.197	0.178	1.100	0.330	4.155
	5	18.934	4.428	11.350	1.192	0.202	1.800	0.246	4.153
	all	22.438	4.390	18.400	1.159	0.240	2.550	0.150	4.302
3-4y	1	16.645	5.446	2.900	1.500	0.182	0.000	0.539	4.984
	3	18.344	5.348	6.450	1.485	0.251	0.100	0.297	5.000
	5	19.668	5.357	9.200	1.493	0.296	0.200	0.211	4.990
	all	23.885	5.270	16.050	1.449	0.360	0.700	0.114	5.127
4-5y	1	27.381	4.943	2.850	1.350	0.260	0.250	0.443	5.096
	3	25.607	4.880	9.000	1.315	0.203	2.000	0.295	5.094
	5	27.058	4.842	14.950	1.306	0.183	3.250	0.223	5.072
	all	31.231	4.778	23.000	1.280	0.191	4.600	0.146	5.300

Table 15: Average metrics by age (where "m" is short for "months" and "y" is short for "years") and number of Student-Teacher Turns (**TurnNo.**; ordered as 1, 3, 5, all). Normalized average uses min–max normalization across model outputs per metric. Note that this table is continued below and on the next page to accommodate all of the linguistic complexity metrics we measured.

Age	TurnNo.	VerbOverlap	AvgClauses	MATTR	AvgFam	Rep.	NumCon	AdversativeCon
6-11m	1	0.152	2.316	0.629	13.563	0.844	4.350	0.500
	3	0.193	2.301	0.583	13.638	0.837	9.300	1.400
	5	0.206	2.373	0.563	13.704	0.815	14.000	2.050
	all	0.230	2.267	0.517	13.756	0.849	20.150	2.800
18-23m	1	0.171	1.833	0.600	13.597	0.659	3.750	0.700
	3	0.188	1.958	0.548	13.639	0.698	9.300	1.900
	5	0.188	1.926	0.534	13.604	0.732	15.500	3.250
	all	0.209	1.902	0.485	13.617	0.766	22.200	5.150
2-3y	1	0.101	2.220	0.691	14.155	0.674	4.800	1.200
	3	0.146	2.244	0.642	14.260	0.734	10.400	2.650
	5	0.175	2.302	0.624	14.329	0.776	16.000	3.500
	all	0.234	2.444	0.561	14.232	0.833	26.500	4.850
3-4y	1	0.146	1.697	0.666	13.255	0.818	3.200	0.450
	3	0.183	1.826	0.593	13.310	0.821	7.350	1.300
	5	0.207	1.959	0.567	13.335	0.838	10.800	1.850
	all	0.238	1.799	0.515	13.271	0.878	20.900	3.500
4-5y	1	0.249	2.811	0.564	13.331	0.650	3.450	0.650
	3	0.159	2.496	0.580	13.736	0.878	11.550	1.700
	5	0.146	2.676	0.589	13.844	0.869	18.800	2.600
	all	0.127	2.684	0.546	13.776	0.858	29.100	3.850

Table 15 (contd.): Average metrics by **Age** (where "m" is short for "months" and "y" is short for "years") and number of Student-Teacher Turns (**TurnNo.**; ordered as 1, 3, 5, all). Normalized average uses min–max normalization across model outputs per metric.

Age	TurnNo.	Polysemy	VerbRep	Narrativity	Norm. Avg
6-11m	1	8.826	0.492	0.000	0.438
	3	8.917	0.632	0.000	0.476
	5	8.741	0.688	0.000	0.556
	all	8.585	0.702	0.000	0.611
18-23m	1	10.066	0.636	-0.000	0.299
	3	10.278	0.706	0.000	0.382
	5	10.090	0.703	0.000	0.443
	all	10.067	0.731	-0.000	0.484
2-3y	1	10.200	0.854	-0.000	0.339
	3	10.706	0.809	-0.000	0.388
	5	10.976	0.803	-0.000	0.457
	all	10.844	0.817	-0.000	0.561
3-4y	1	7.821	0.562	-0.000	0.354
	3	7.651	0.617	0.000	0.393
	5	7.514	0.637	0.000	0.437
	all	7.403	0.669	0.000	0.504
4-5y	1	8.411	0.629	-0.000	0.398
	3	8.819	0.726	0.000	0.489
	5	9.207	0.681	-0.000	0.526
	all	9.084	0.662	-0.000	0.573

Table 15 (contd.): Average metrics by age (where "m" is short for "months" and "y" is short for "years") and number of Student-Teacher Turns (**TurnNo.**; ordered as 1, 3, 5, all). Normalized average uses min–max normalization across model outputs per metric.

F Analysis of Reward Model Training Dynamics for CPO/ORPO and CEFR Models

F.1 Comparison of Training Reward Dynamics Across Reward Types for CPO/ORPO (Experiment 1) and CEFR Models (Experiment 2)

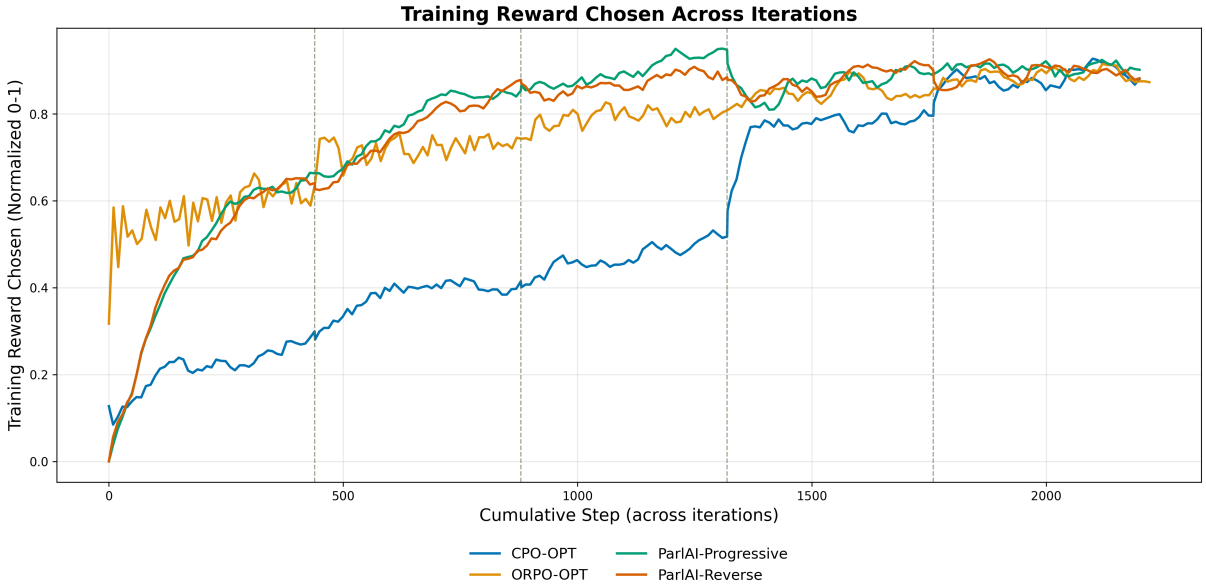


Figure 7: CONTINGENTCHAT Training Reward Chosen for CPO/ORPO Models (Experiment 1) and CEFR Models (Experiment 2)

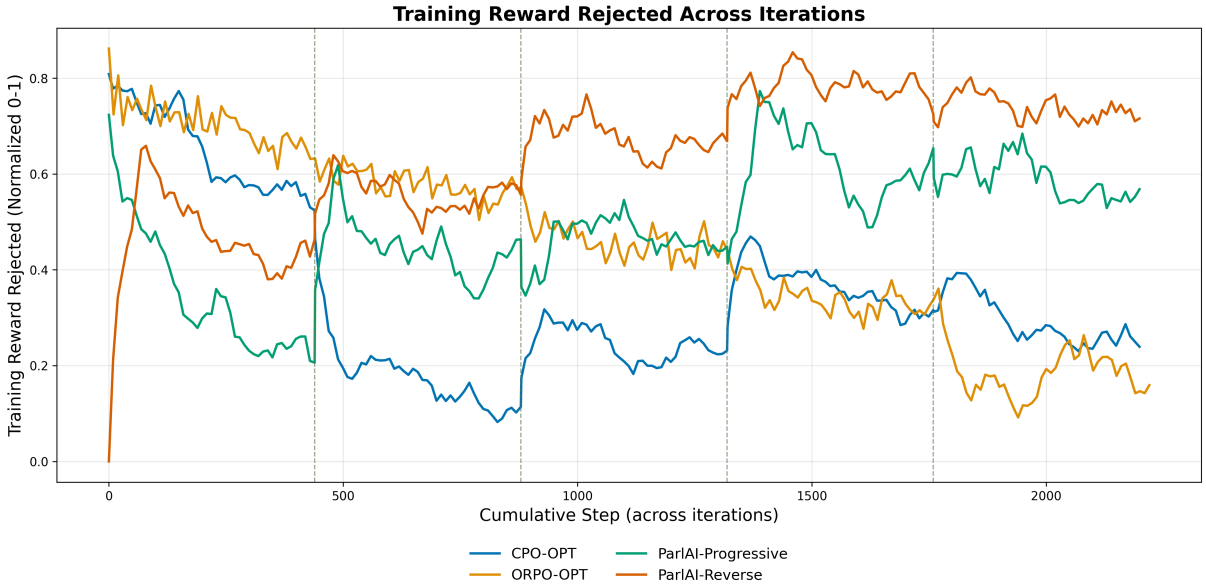


Figure 8: CONTINGENTCHAT Training Rejection for CPO/ORPO Models (Experiment 1) and CEFR Models (Experiment 2)

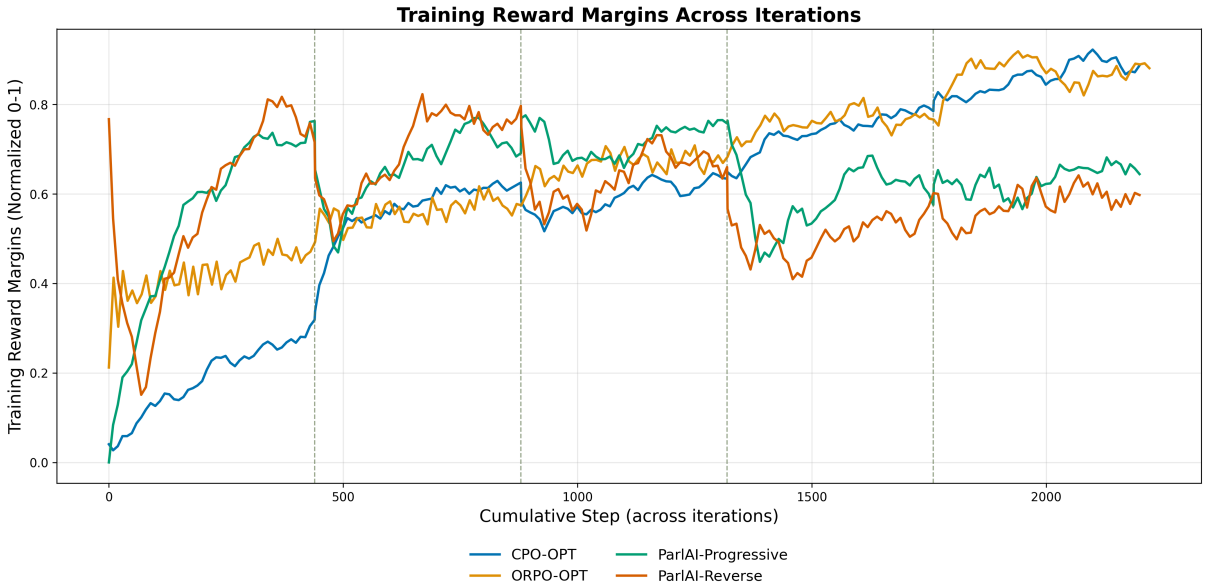


Figure 9: CONTINGENTCHAT Training Reward Margin for CPO/ORPO Models (Experiment 1) and CEFR Models (Experiment 2)

F.2 Progressive CEFR Model Training Reward Dynamics Across Iterations and Reward Types (Experiment 2)

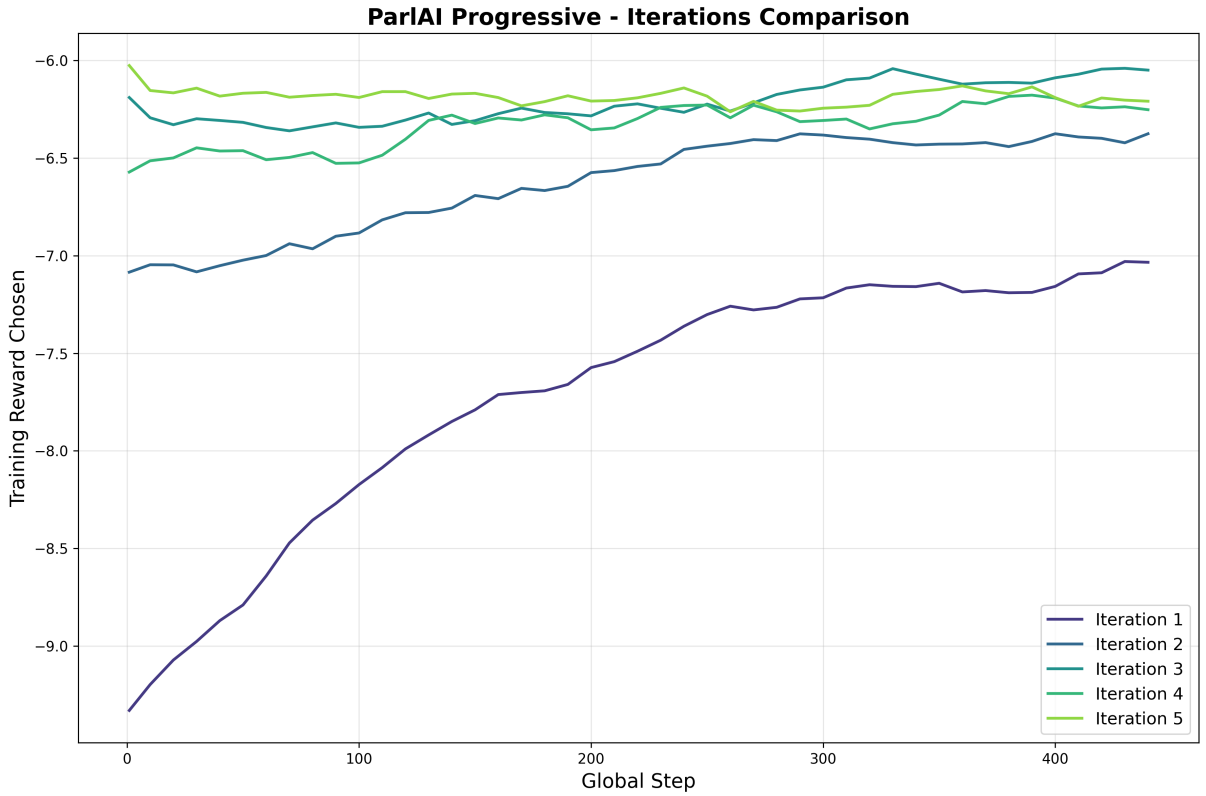


Figure 10: CONTINGENTCHAT Training Reward Chosen of Progressive CEFR model across iterations

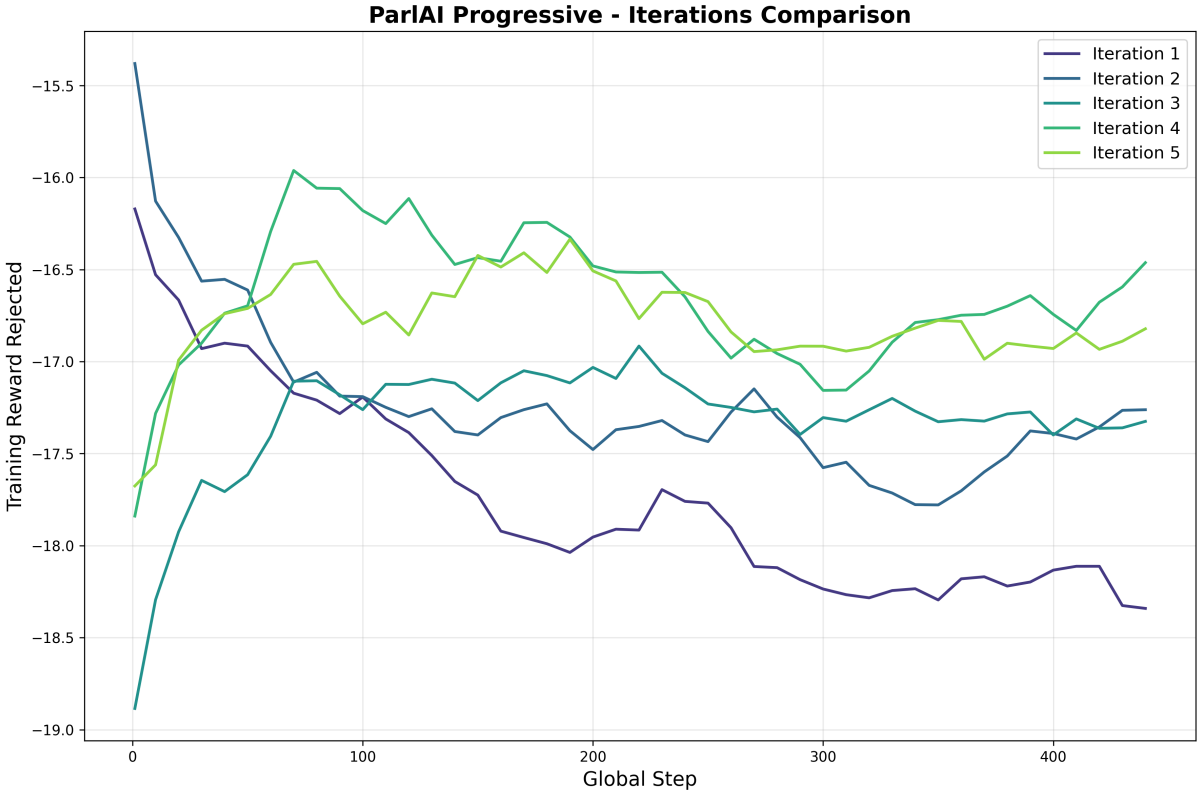


Figure 11: CONTINGENTCHAT Training Reward Rejected of Progressive CEFR model across iterations

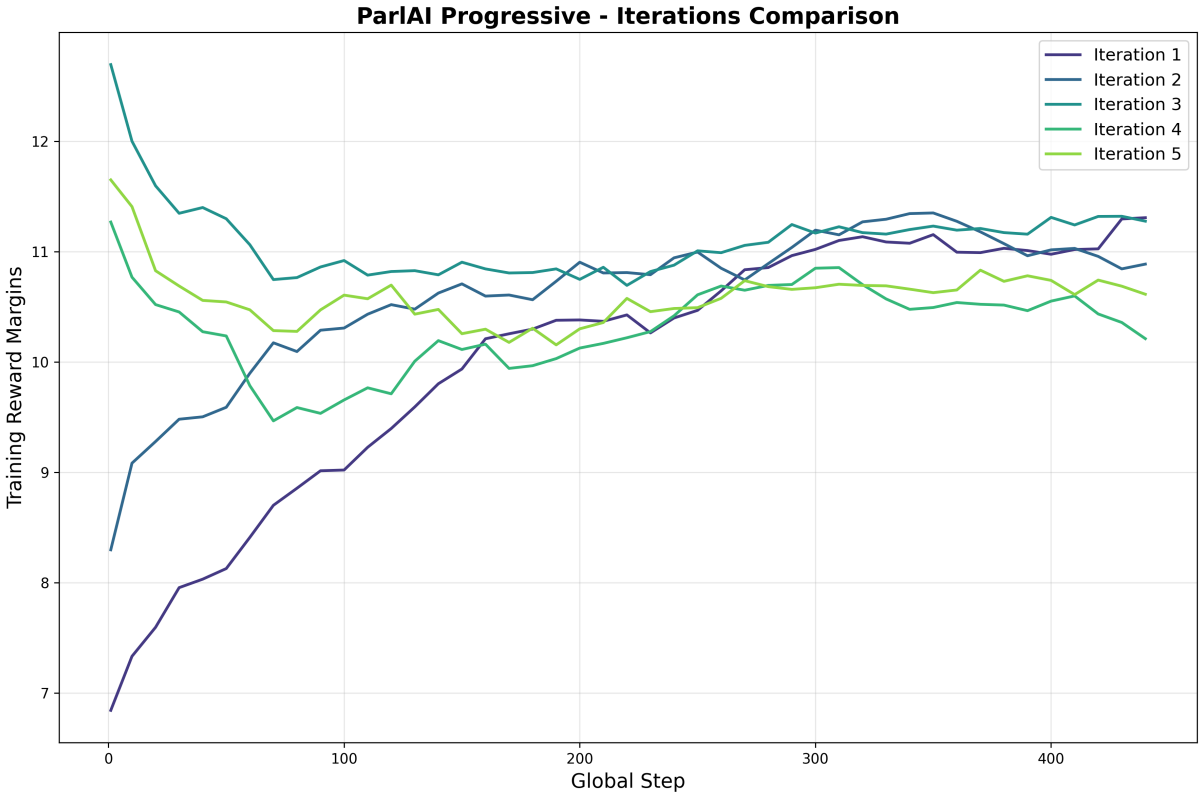


Figure 12: CONTINGENTCHAT Training Reward Margins of Progressive CEFR model across iterations

F.3 Reverse CEFR Model Training Reward Dynamics Across Iterations and Reward Types (Experiment 2)

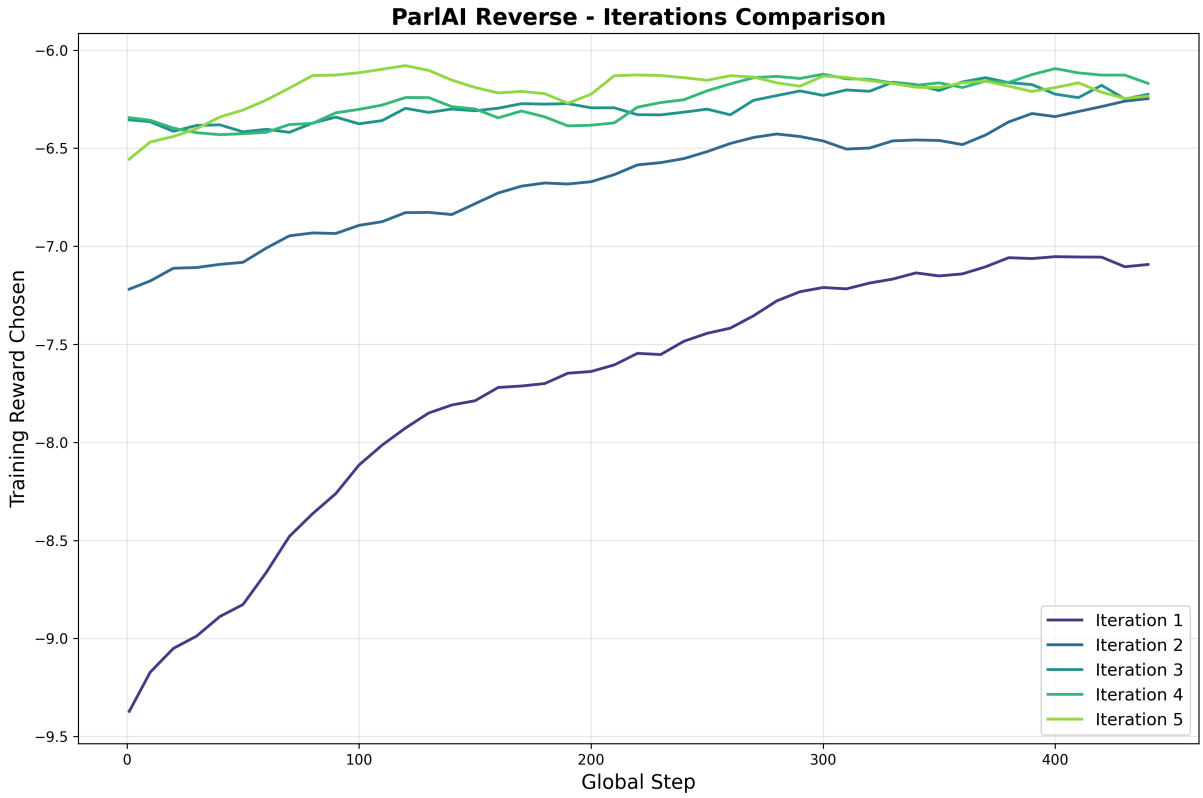


Figure 13: CONTINGENTCHAT Training Reward Chosen of Reverse CEFR model across iterations

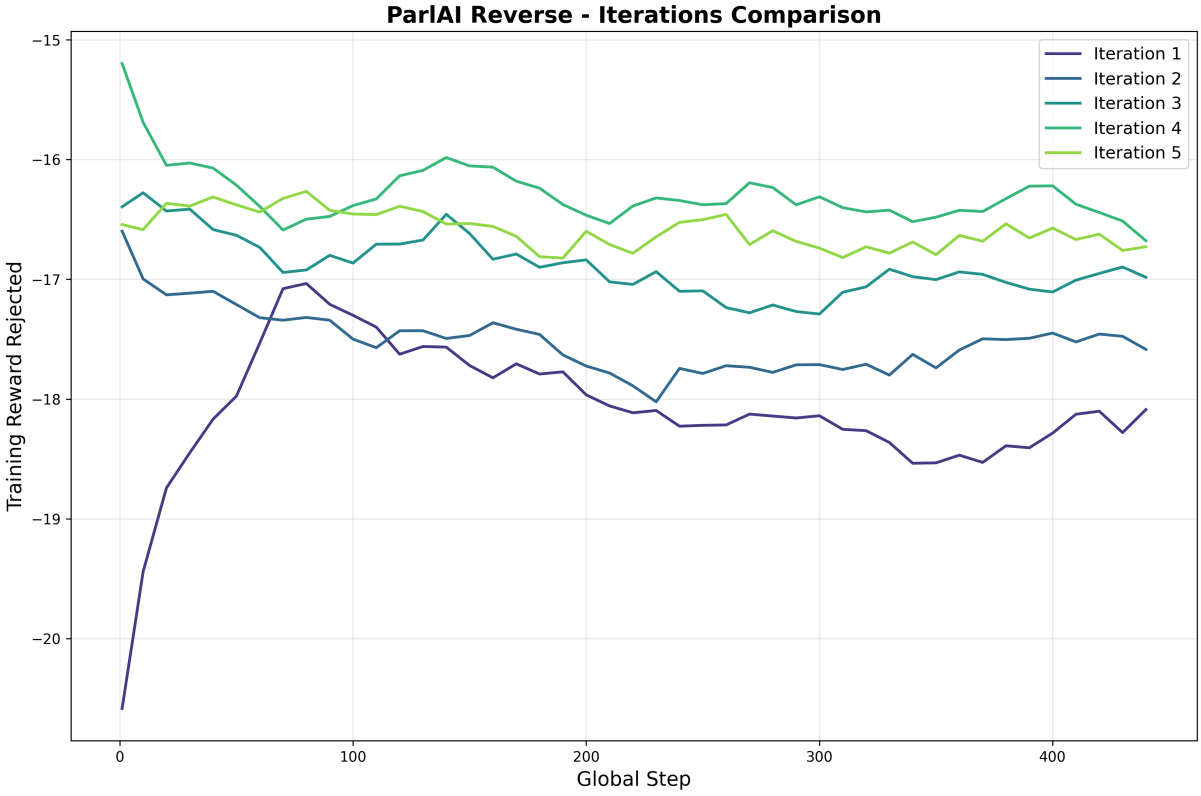


Figure 14: CONTINGENTCHAT Training Reward Rejected of reverse CEFR model across iterations

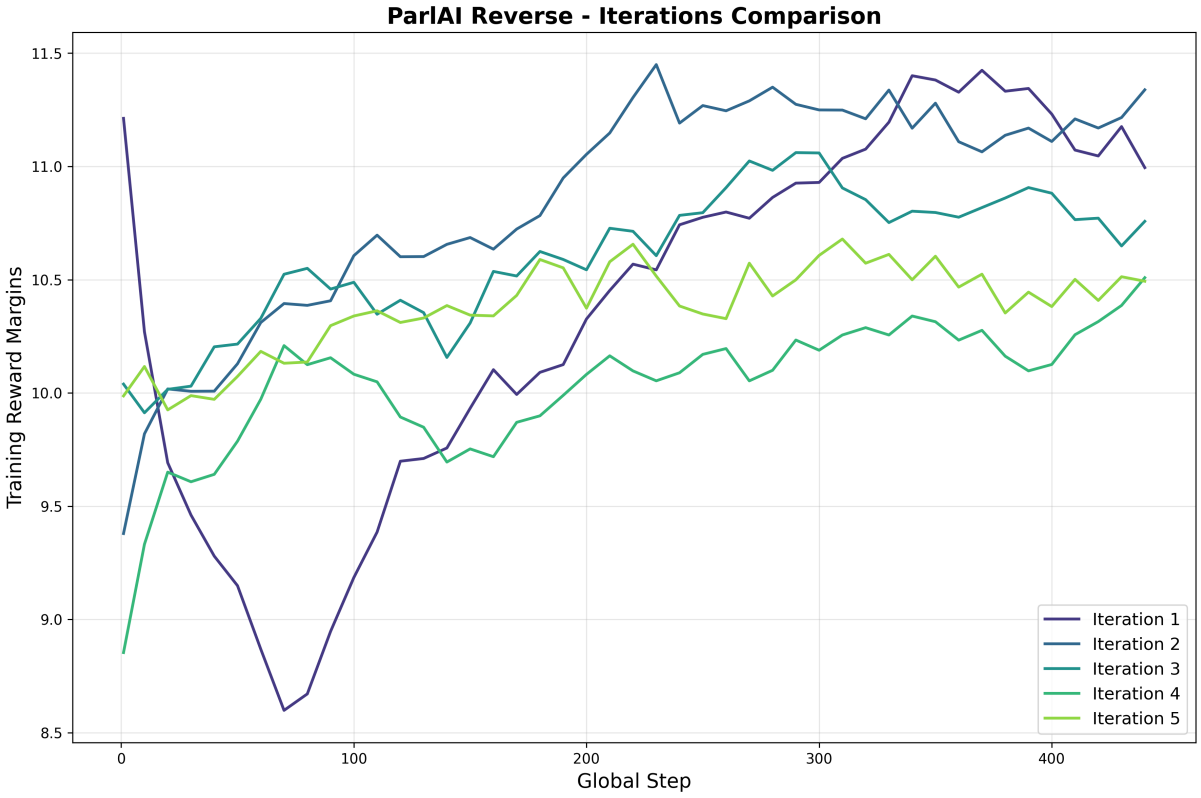


Figure 15: CONTINGENTCHAT Training Reward Margins of reverse CEFR model across iterations

Influence-driven Curriculum Learning for Pre-training on Limited Data

Loris Schoenegger[1,2], **Lukas Thoma**[1,2],
Terra Blevins[1,4], **Benjamin Roth**[1,3]

[1]Faculty of Computer Science, University of Vienna, Vienna, Austria
[2]UniVie Doctoral School Computer Science, University of Vienna, Vienna, Austria
[3]Faculty of Philological and Cultural Studies, University of Vienna, Vienna, Austria
[4]Khoury College of Computer Sciences, Northeastern University, Boston, USA

Correspondence: loris.schoenegger@univie.ac.at

Abstract

Curriculum learning, a training technique where data is presented to the model in order of example difficulty (e.g., from simpler to more complex documents), has shown limited success for pre-training language models. In this work, we investigate whether curriculum learning becomes competitive if we replace conventional human-centered difficulty metrics with one that more closely corresponds to example difficulty as observed during model training. Specifically, we experiment with sorting training examples by their *training data influence*, a score which estimates the effect of individual training examples on the model's output. Models trained on our curricula are able to outperform ones trained in random order by over 10 percentage points in benchmarks, confirming that curriculum learning is beneficial for language model pre-training, as long as a more model-centric notion of difficulty is adopted.

1 Introduction

Curriculum learning, a training paradigm where the training data is presented to the model in non-random order (Bengio et al., 2009), has recently been explored extensively as a pretraining strategy for language models due to its potential to improve performance in low-resource settings (Timiryasov and Tastet, 2023), reduce training time (Platanios et al., 2019), or to make the training process more data-efficient and developmentally plausible (i.e., more similar to how humans acquire language; Warstadt et al., 2023a; Hu et al., 2024). A popular form of curriculum learning relies on heuristics that **sort training data by increasing difficulty** (e.g., lexical diversity trough type-token ratio: Mi, 2023). However, in low-resource language modeling, approaches that incorporate this curriculum learning strategy have not yielded the anticipated improvements and show no consistent positive effect on model performance (Hu et al., 2024). In this work, we therefore investigate whether curriculum learning becomes competitive for language model pretraining, if we replace human-centered difficulty measures with one that better reflects training dynamics. Specifically, we derive a novel form of curriculum from **training data influence estimates**, that we obtain from a surrogate model trained with randomly ordered data: These estimates assign documents from the training data scores proportional to their impact on the model's output. We adapt a *gradient similarity-based* influence score (Pruthi et al., 2020), where influence is measured by comparing loss-gradients of training and test instances, with higher similarity signifying greater influence. We experiment with 10 different sorting

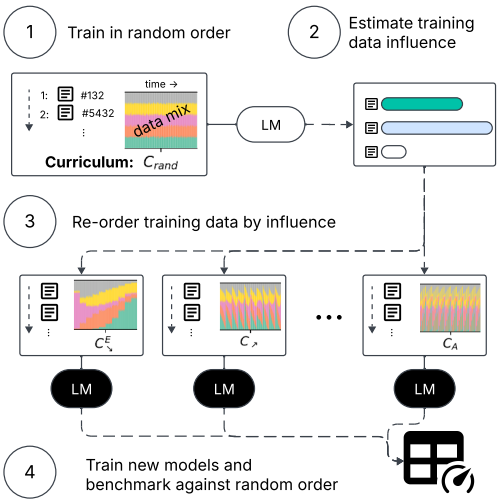


Figure 1: In our method, we extract training data influence estimates from models trained in random order, to create better-performing curricula.

strategies, all based on the **average influence** that a given training example exerts on the prediction of *other* examples sampled from the training data. We compare model performance under these curricula to both random training and curriculum learning using three human-centered difficulty heuristics. Through experiments with RoBERTa- (Liu et al.,

Proceedings of the First BabyLM Workshop, pages 368–391

2019) and Llama models (Touvron et al., 2023), we demonstrate that our approach is more effective than handcrafted curricula, and analyze what ranking and coverage strategies are most effective. We find that source-difficulty curricula, a popular human-centered design that arranges datasets by their difficulty, are ineffective compared to alternative dataset coverage strategies, and we offer insights into the reasons for their low performance. Our **main contributions** are as follows:[1]

(1) We demonstrate that our curricula yield an increase of over 10 percentage points (pp) in accuracy for RoBERTa- and over 4 pp for Llama models on a popular challenge dataset for low-resource pre-training (BabyLM 10M-word dataset: Choshen et al., 2024).
(2) We analyze the data mix of the generated curricula (e.g., child-directed speech, dialogue, etc.) and how it evolves over time;
(3) Analyze loss trajectories to study how our curricula affect the model's learning process;
(4) Explore how example ordering within influence curricula relates to existing heuristics.

2 Related Work

Curriculum Learning can roughly be categorized into dynamic and static approaches. Dynamic designs incorporate difficulty heuristics directly into the training process, generating or updating the curriculum during training (e.g., Kumar et al., 2010; Sedova et al., 2023). Static curricula have recently proven popular in the *BabyLM challenge*, a competition promoting the creation of more developmentally plausible language models (Hu et al., 2024): Motivated by the observation that humans only require up to 100 million words to reach native levels in a language (Gilkerson et al., 2017), this challenge invites NLP researchers to explore human-centered learning strategies on a dataset of just 10M or 100M words. Participants have incorporated various sorting heuristics into curriculum learning schemes, such as sorting by increasing sentence length (Platanios et al., 2019; Ghanizadeh and Dousti, 2024; Borazjanizadeh, 2023; Spitkovsky et al., 2010), document- or sentence complexity (Oba et al., 2023; Opper et al., 2023), lexical diversity (Mi, 2023; Ghanizadeh and Dousti, 2024), or dataset-level source difficulty by category (Thoma et al., 2023; Huebner et al., 2021; Martinez et al., 2023; Opper et al., 2023). However, static approaches following this framework have shown no consistent positive effect on model performance (Hu et al., 2024).

Our method is motivated by the assumption that children's language learning proceeds from easy to complex input (Elman, 1993), but represents a middle ground between static and dynamic approaches: we generate static curricula, but base them on a score that reflects training dynamics.

Training Data Influence for CL Bejan et al. (2023) employ TracIn self-influence (Pruthi et al., 2020) for curriculum learning in the fine-tuning setting. For them, self-influence is defined as $\nabla\ell(w_t, z) \cdot \nabla\ell(w_t, z)$ (Pruthi et al., 2020), which does not relate to other data points in the training data, and effectively only quantifies magnitude for a given example. In contrast to our approach, their focus lies on improving performance by filtering outliers and up-weighting the most influential examples. Our approach incorporates more information, specifically pairwise influence scores between one example and *all* other examples in the training data, as outlined in Section 3.1.

Role of Example Difficulty in Learning Several authors have utilized measures of example difficulty to systematically study the effect of curriculum learning for supervised fine-tuning tasks and in the image domain (Hacohen and Weinshall, 2019; Wu et al., 2020; Jiang et al., 2021; Baldock et al., 2021). For instance, Wu et al. (2020), study whether examples of similar difficulty are learned at similar stages across architectures through comparing the *learned iteration* of examples across models, a metric defined as the first epoch at which the model correctly predicts them. Our setup differs in that we study the model's downstream performance and operate within an unsupervised setting.

3 Methodology

In this work, we investigate the benefits of incorporating training data influence estimates into curriculum learning methods, particularly for low-data pretraining settings. We first introduce our approach for estimating example difficulty using training gradients. Then, we describe our curriculum designs and outline our experimental setup.

[1]We release our code at https://doi.org/10.5281/zenodo.16919045, and host all datasets and models on the Hugging Face Hub at https://huggingface.co/collections/loris3/ticl-68a6fd8bcc3093f239439e42.

3.1 Training Data Influence Estimation

We define a new metric for measuring example difficulty in curriculum design that leverages training data influence estimates: We adapt TracinCP (Pruthi et al., 2020) for this, which in its original formulation estimates the *point-wise influence* $\phi_{\text{TracInCP}}(z, z')$ that training on an instance z had on the model, when predicting a test instance z'. The estimation process involves measuring the similarity between the gradients of the model's loss function, when evaluated on z and z' respectively, w.r.t some set of parameters w_t, and is repeated at a series of checkpoints T:

$$\phi_{\text{TracInCP}}(z, z') = \sum_{\forall t \in T} \eta_t \nabla \ell(w_t, z) \cdot \nabla \ell(w_t, z') \quad (1)$$

Following Yeh et al. (2022), we let w_t be the model's input embeddings at checkpoint t.[2] To leverage this point-wise influence score **for curriculum learning**, we propose to calculate the **average influence** $\phi_t(z, D)$ that a given training example exerts on the prediction of all other examples from the training data D. Omitting the learning rate η_t, for one training instance z, and one checkpoint t we calculate:

$$\phi_t(z, D) = \frac{\sum_{\forall z' \in D} \nabla \ell(w_t, z) \cdot \nabla \ell(w_t, z')}{|D|} \quad (2)$$

$$= \nabla \ell(w_t, z) \cdot \mathbb{E}_{z' \sim D}[\nabla \ell(w_t, z')] \quad (3)$$

Intuitively, this score quantifies the average utility of a given example during training. Unlike measures of surprisal such as perplexity, it is high for prototypical examples (which feature loss gradients similar to the average gradient) and low for outliers. Doing so for all examples in the training dataset D, at regular checkpoints for a model trained in random order, yields a matrix $\Phi \in \mathbb{R}^{|D| \times |T|}$ like the one depicted in Figure 2, which we subsequently use for constructing curricula with various reordering functions. In initial experiments, we observed that this score based on dot-product similarity was biased against longer examples, which was also observed by Xia et al. (2024). Thus, **we normalize the loss gradients** to reduce the impact of gradient magnitude on the similarity scores, effectively yielding cosine similarity (Hammoudeh and Lowd, 2022, 2024; Park et al., 2023; Xia et al., 2024).

[2]Note that this score incorporates information about the full model, as the gradient chains through higher layers as well (Yeh et al., 2022).

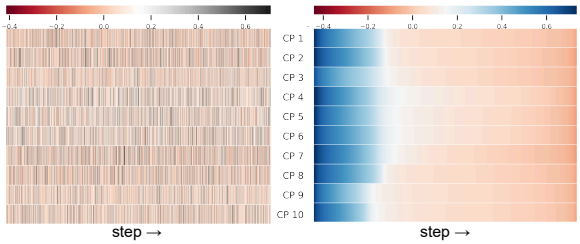


Figure 2: Left: measured influence on C_{rand}; Right: anticipated influence if sorted according to $C_{\searrow}$.

3.2 Curriculum Design

This section introduces our 10 curriculum design methods based on influence estimates, as well as 4 baseline curricula. Our designs can be broadly categorized into two categories, characterized by their coverage strategy: the first group of curricula covers the full dataset every epoch, while the second group progressively increases example difficulty across epochs, consequently not re-visiting examples from early epochs in later ones.

Epoch-wise Dataset Coverage Strategies

In the curricula $C_{\searrow}$ and $C_{\nearrow}$, we sort documents in descending ($\searrow$) or ascending ($\nearrow$) order of influence, measured using model checkpoints of a surrogate model trained in random order stored after each epoch t. We include an additional pair of curricula $C^{\sim}_{\searrow}$ and $C^{\sim}_{\nearrow}$, where, in an attempt to increase data diversity during training, we additionally divide the curriculum into ordered subsets of 1000 documents, and then randomly shuffle the documents within these subsets. Similarly, motivated by the intuition that examples with lasting influence across epochs should be prioritized because they appear to have been more difficult for the surrogate model to learn, we add a re-weighting step to the two curricula $(C * h)^{\sim}_{\searrow}$ and $(C * h)^{\sim}_{\nearrow}$, where we convolve the influence estimates Φ with a lognormal filter h before the sorting step; this thus upweights examples that remain influential in subsequent epochs: $(C * h)_{(t,i)} = \sum_{k=0}^{T} \Phi_{(t-k,i)} \cdot \mathrm{h}(k)$.

Lastly, emulating prior works that used influence estimates for data cleaning and not solely for re-ordering (e.g., Bejan et al., 2023), we add a curriculum $C^{\{50\}}$ where we discard the 50% least influential examples in each epoch, while keeping the total number of words shown to the model constant. We shuffle once per epoch.

Cumulative Dataset Coverage Strategies

Source difficulty curricula (Martinez et al., 2023) are a curriculum learning strategy where models are

trained on a collection of datasets that are manually sorted by difficulty (but the individual examples within these datasets are not). In $C^E_{\searrow}$ and $C^E_{\nearrow}$, we design a similar coverage strategy, allowing us to subsequently test whether curricula based on training data influence yield similar dataset mixtures as handcrafted ones: In contrast to the curriculum designs introduced so far, we aggregate the individual influence estimates for a given example across all T epochs to obtain a measure of its overall influence during training ($\phi_T(z, D) = \sum_{\forall t} \phi_t(z, D)$). We then sort examples by this score, either in ascending ($\nearrow$) or descending order ($\searrow$). Subsequently, we divide this ordered data into $m = 10$ segments, from which we then randomly sample to create m equal-length epochs with examples of increasing or decreasing difficulty respectively.

Our last curriculum, C_A, is designed as a compromise between curricula with epoch-wise dataset coverage strategies and $C^E_{\cdot}$: In this curriculum, we alternate between showing subsets of high influence scores and subsets of low influence scores, but shuffle the individual examples within each segment randomly. Specifically, we first sort examples by their aggregate score $\phi_T(z, D)$, and create $m = 10$ segments just as for $C^E_{\cdot}$. We then assemble the curriculum from these segments by alternating between the highest-influence and lowest-influence ones until all are used. We train for 10 epochs in this order, randomly shuffling the examples within each segment before each pass.

Baseline Curricula

We include 4 baseline curricula C_{rand}, C_{source}, C_{MATTR} and C_{PPL}: In C_{rand} we emulate non-curriculum learning, performing 10 full passes over the training data in random order. We train one model per dataset using this curriculum, storing checkpoints after each full pass so that it can serve both as a surrogate model for extracting influence estimates and as a baseline (we utilize a total of $T = 10$ checkpoints).

Handcrafted *source-difficulty curricula* present datasets sorted by difficulty as distinct blocks (e.g., children's books before Wikipedia articles). We define such a curriculum in C_{source}, by assigning the datasets in Table 1 to one of 5 stages (C1-C5), following previous work (Thoma et al., 2023; Huebner et al., 2021; Martinez et al., 2023; Opper et al., 2023). Similar to $C^E_{\nearrow}$ and $C^E_{\searrow}$, we train for two epochs per stage, randomly shuffling examples within each epoch.

C_{MATTR} is inspired by Mi's (2023) use of *type-token ratio* (TTR) for curriculum learning. Here, we sort documents by increasing moving average type-token ratio (C_{MATTR} (with a window length of 5); Covington et al., 2010).[3] Lastly, for C_{PPL}, we sort in order of increasing perplexity under a static uni-gram model, as described in Martinez et al. (2023). With both C_{MATTR} and C_{PPL}, we train the model on full epochs in this order 10 times.

3.3 Datasets

We train models on three datasets:

- D_{2024} is the 10M word text-only dataset utilized in the 2024 and 2025 iterations of the BabyLM challenge (Choshen et al., 2024; Charpentier et al., 2025), which is composed of datasets of various levels of difficulty listed in Table 1.
- To facilitate analysis of source-difficulty curricula, we construct $D_{stratified}$, which has an equal number of words per stage. We sample from the same datasets underlying D_{2024}, but add sources to balance word counts (Table 1).
- As document length varies substantially by source, we additionally control for the number of words per document in a third dataset $D_{equitoken}$ (also stratified and balanced w.r.t stages); specifically, we create synthetic documents that are exactly 100 words long by concatenation.

Finally, we create a shared evaluation set for all D_*, sampled from the 100M word version of said BabyLM dataset ($|D_{eval}| = 0.05 \cdot |D_{2024}|$).

3.4 Models

Our experiments produce a total of 84 models, one RoBERTa- (126M params) and one Llama model (97.2M params), both with random initializations, for each combination of the 3 datasets and 14 curricula. We train on 4 NVIDIA H100 GPUs with an effective batch size of 2048, using the parameters summarized in Table 3 in Appendix A. Each curriculum includes at most 100 million words (e.g., 10 passes over a dataset of 10M tokens for C_{rand}).

4 Results and Analysis

This section presents and analyzes the results of our curriculum design experiments. Specifically, we:

[3] We choose to use MATTR over TTR as a metric to make our curricula more robust to variation in document length.

C1	**Child Directed Speech** CHILDES (MacWhinney, 2014)
C2	**Unscripted Dialogue** Switchboard Dialog Act Corpus (Stolcke et al., 2000) British National Corpus (BNC), dialogue portion (Consortium, 2007)
C3	**Scripted Dialogue** OpenSubtitles (Lison and Tiedemann, 2016)
C4	**Wiki** Simple Wiki (Warstadt et al., 2023a)
C5	**Written English** Standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2018)

C1	**Child Directed Speech** CHILDES (MacWhinney, 2014)
C2	**Children's Books** Children Stories Text Corpus (Bensaid et al., 2021) Children's Book Test (Hill et al., 2016)
C3	**Dialogue** Switchboard Dialog Act Corpus (Stolcke et al., 2000) British National Corpus (BNC), dialogue portion (Consortium, 2007) OpenSubtitles (Lison and Tiedemann, 2016)
C4	**Educational** Simple Wiki (Warstadt et al., 2023a) QED (Abdelali et al., 2014)
C5	**Written English** Standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2018) Wikipedia (Warstadt et al., 2023a)

Table 1: Curriculum stages in C_{source}. Stages for D_{2024} (left) differ from those in $D_{stratified}$ and $D_{equitoken}$ (right) to allow for a balanced split. We make all three datasets available under CC BY 4.0.

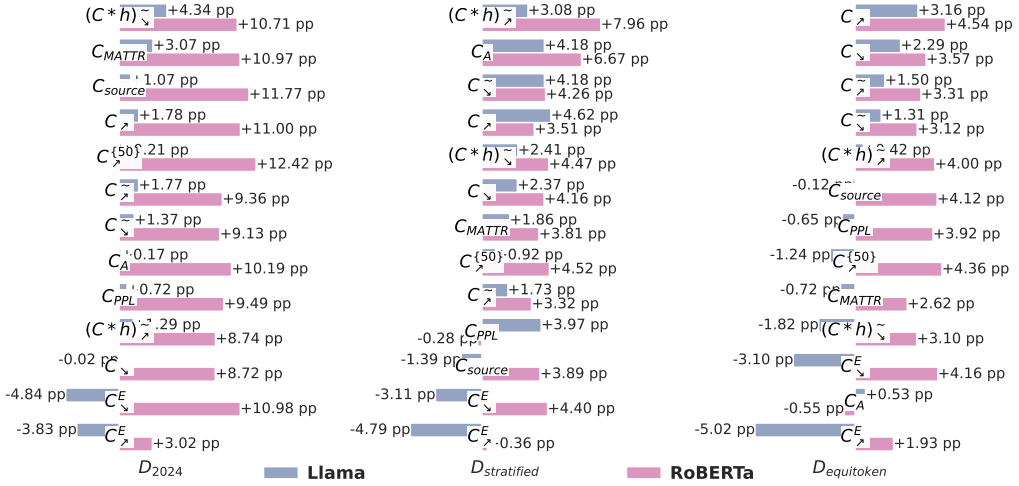


Figure 3: Average change in macro-accuracy across benchmark tasks w.r.t. training on the random curriculum. Sorted by average change across RoBERTa and Llama models.

(1) present the benchmark performance of our models on downstream tasks; (2) compare the source composition of our curricula to those of the baselines; (3) analyze training- and evaluation loss trajectories; (4) and explore how example ordering in the influence curricula correlates with the orderings of existing heuristics.

4.1 Benchmark Performance

Dataset	Llama C_{rand}	Llama $(C*h)^{\sim}_{\nearrow}$	RoBERTa C_{rand}	RoBERTa $(C*h)^{\sim}_{\nearrow}$
D_{2024}	0.541	**0.554**	0.466	0.553
$D_{stratified}$	0.536	0.566	0.512	**0.592**
$D_{equitoken}$	0.523	0.527	0.492	**0.532**

Table 2: Macro-Accuracy across tasks for random order and the $(C*h)^{\sim}_{\nearrow}$ curriculum.

We evaluate our curricula by comparing their performance to models trained on the same data in random order. In Figure 3, we report the increase or decrease in macro-accuracy across individual benchmark tasks from BLiMP (Warstadt et al., 2020), BLiMP supplement (Warstadt et al., 2023b), EWOK (Ivanova et al., 2024), Super GLUE (Wang et al., 2019), as well as an entity tracking task (Kim and Schuster, 2023) and an adjective nominalization task (Hofmann et al., 2024), as implemented specifically for the BabyLM challenge (Charpentier et al., 2025). Results for the individual benchmarks are provided in Appendix D.

In terms of raw performance, the RoBERTa model trained using $(C*h)^{\sim}_{\nearrow}$ (sorted by increasing influence, re-weighted with lognormal filter) on $D_{stratified}$ is the **best performing model overall** (0.592 macro-acc, +7.96 pp over C_{rand}), with the best Llama model being the one trained with $(C*h)^{\sim}_{\searrow}$ (sorted by decreasing influence) on D_{2024} (0.584, +4.34 pp). RoBERTa models see higher absolute gains through the addition of curriculum learning than Llama models in our experiments. This can partially be attributed to their lower initial accuracy when trained in random order, with Llama models outperforming RoBERTa models by 7.5, 2.4, and 3.1 pp on the D_{2024}, $D_{stratified}$,

and $D_{equitoken}$ datasets, respectively (Table 2). Notably, for RoBERTa models, the handcrafted source curriculum was effective on D_{2024} (+11.77 pp), and only two curricula lead to a decrease in performance, namely C_{PPL} on $D_{stratified}$ (-0.28 pp), and C_A on $D_{equitoken}$ (-0.55 pp). For Llama models, in contrast, the worst-performing curricula $C^E_{\searrow}$ and $C^E_{\nearrow}$ incur a considerable 3.10-5.02 pp decrease in accuracy over training in random order.

For both model architectures, the highest gains through curriculum learning are on D_{2024} followed by $D_{stratified}$ (equal number of words per stage), and $D_{equitoken}$ (equal number of documents per stage, and words per document).

Dataset Coverage Strategies
Models trained with handcrafted- (C_{source}) and synthetic source difficulty curricula ($C^E_{\searrow}$, $C^E_{\nearrow}$), both designed to increase difficulty gradually across epochs (cumulative coverage strategies), perform worse overall than the other designs, which perform one full pass over the data each epoch (per-epoch coverage strategies). C_A, where we alternate between showing subsets of high influence scores and subsets of low influence scores, shows significant improvements over training in random order for both Llama (+4.18 pp) and RoBERTa (+6.67 pp) on $D_{stratified}$ and D_{2024} for RoBERTa (+10.19 pp), but not for the remaining three models.

Sorting Direction and Shuffling Strategy
Surprisingly, our benchmark results do not conclusively show whether curricula sorted by ascending ($\nearrow$) or descending ($\searrow$) influence perform better; the ascending version of the same strategy does not consistently outperform the descending version (and vice versa). Curricula where we shuffle within stages (e.g., $C^{\sim}_{\searrow}$) similarly do not reliably outperform ones without, the same applies to curricula built from lognorm-filtered influence estimates ($(C * h)^{\sim}_{\searrow}$). We offer a potential explanation for this in Section 5.

4.2 Source Composition

The datasets we utilize are themselves composed of sources of varying difficulty; similar to previous work (Thoma et al., 2023) we have attributed each to one of five stages of increasing difficulty (C1-C5; from a human learning perspective) for constructing the handcrafted curricula (Table 1). Based on these labels, we plot the source compositions of the training data shown to the Llama models over time in Figure 4 and provide those of RoBERTa models in Appendix C.

We observe that our **influence curricula are highly sensitive to the source distribution of the dataset**. *C1: Child Directed Speech* and *C3: Dialogue*, the two largest stages in the unbalanced D_{2024} dataset, are scheduled first in the synthetic source difficulty curriculum $C^E_{\searrow}$, with more than half of the training steps allocated to them. For $C^{\{50\}}$, where we discard the 50% least influential examples in each epoch, the share of child directed speech accounts for over 90% of examples throughout the training process, despite accounting for only roughly half of D_{2024} by number of documents.

This **over-representation of child directed speech** in the majority of epochs may explain why these curricula perform worse in benchmark tasks than all other influence curricula across all datasets and model types: When controlling for the number of words per source ($D_{stratified}$), the effect is less extreme, yet, *C1: Child-Directed Speech*, *C3: Dialogue*, and *C4: Educational* are more frequently shown in early rather than in later epochs in $C^E_{\searrow}$, with *C5: Written English* following the opposite trend. For $D_{equitoken}$ however, where the model used for influence estimation sees an equal number of tokens and documents per stage, all trends are reversed, with C1 now shown more often in later epochs, and C5 in earlier ones. One possible explanation stems from the definition of our datasets, which sample based on a word-based budget rather than one based on the number of documents: In D_{2024}, C1 accounts for 54% of documents but only 28% of words, while C5 comprises 25% of words within just 6% of the dataset's documents.[4] Because our sorting relies on a *per-document average* influence measure, similarity to the larger subset C1 likely disproportionately impacts influence scores compared to similarity with C5. This suggests that our ranking method is **biased against smaller sources** (by number of documents).

Contrary to our initial expectation that the influence of child-directed speech would diminish in later epochs, the **source composition** of epoch-wise dataset coverage strategies (e.g., $C_{\searrow}$), **does not strongly vary over time**. To obtain a formal measure of how similar a curriculum's source distribution over time is to the model-agnostic baselines, we split both curricula into $n = 1000$ segments, for which we then calculate the average

[4] the same pattern applies in $D_{stratified}$

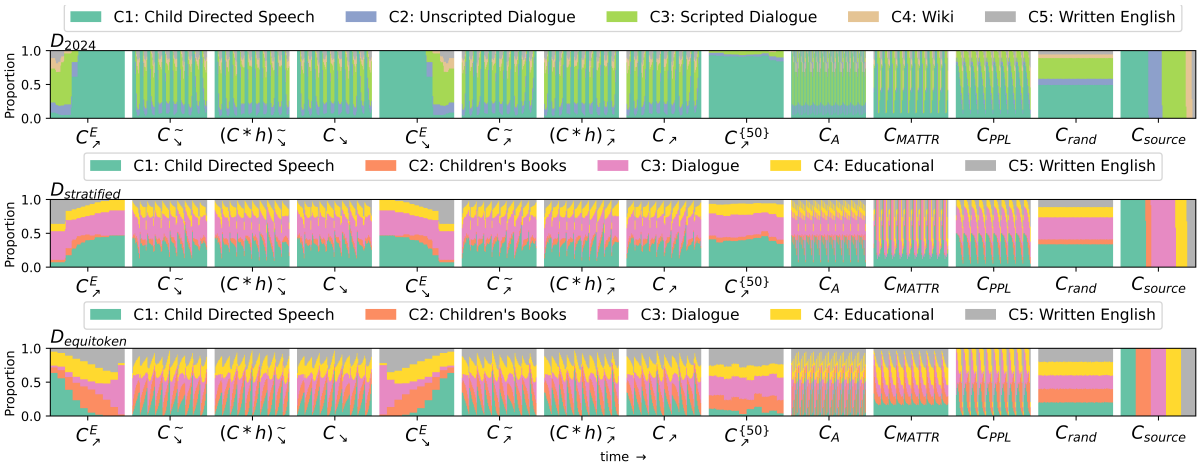


Figure 4: Dataset mix of curricula for Llama models. We trace back documents to the stages defined in Table 1.

Jensen-Shannon divergence[5]. We find that our curricula's source distribution is closer to that of C_{rand} than to other baselines (i.e., our curricula retain the dataset's source distribution, Figure 5). We therefore cannot explain the performance of influence curricula through their source distributions alone.

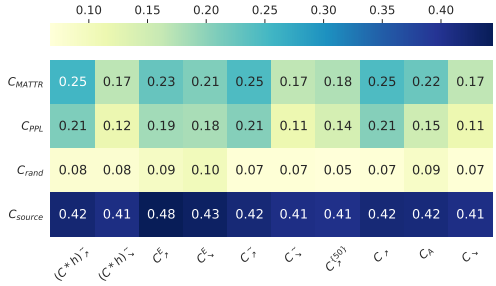


Figure 5: Average Jensen-Shannon divergence between curricula for Llama models. Lower values indicate more similar stage distributions.

4.3 Loss Trajectories

We provide training- and evaluation loss trajectories for a subset of our models in Figure 6, and the remaining ones in Appendix D. For one RoBERTa model ($C^E_{\nearrow}$ on D_{2024}) and 9 Llama models (D_{2024}: $\{C^E_{\nearrow}, C_A, C_{source}, C_{MATTR}, C_{rand}\}$, $D_{stratified}$: $\{C^E_{\nearrow}, C^E_{\searrow}, C^{\{50\}}_{\nearrow}, C_{rand}\}$) we measure higher evaluation loss at the end of training compared to the beginning, suggesting training divergence.

We observe **substantial training loss spikes**, which in non-curriculum learning often indicates training instability (Li et al., 2022). However, as evident in Figure 6, the model that performs best in benchmarks ($(C*h)^{\sim}_{\nearrow}$, RoBERTa) exhibits more severe training loss-spikes than the worse performing C_{source}, $C^E_{\searrow}$ or C_{random}. We extend this analysis to all 84 models, calculating the Spearman rank correlation between a curriculum's gain in benchmark performance (over training in random order) and the *loss-ratio* (a measure of training instability; Li et al., 2022) in Appendix B. We find no significant negative rank correlation for any dataset[6], indicating that at least within the limited number of epochs we train for, training loss trajectories appear **less informative of downstream performance** compared to training in random order.

4.4 Document Order

We additionally explore how the ordering of examples under influence curricula correlates with ordering of existing heuristics. We use Kendall's τ, calculated on a per-epoch basis as documents are shown multiple times during training.[7] Curricula sorted by decreasing influence ($C^{\sim}_{\searrow}$, $C_{\searrow}$, $(C*h)^{\sim}_{\searrow}$) show significantly stronger correlations with both C_{MATTR} and C_{PPL} than curricula sorted by increasing influence (C_{MATTR} : $+0.047^*$, C_{PPL} : $+0.084^*$). This suggests that our influence measure **may be inversely related to example difficulty** as defined by these curricula (i.e., higher influence implies lower difficulty). Rank correlation between any type of influence curriculum and C_{rand}, as well as between influence curricula and C_{source} is negligible, which is to be expected as we shuffle these within epochs or stages respectively. Convolving

[5] $\mu_{JSD}(p_a||p_b) := \sum_{i=1}^{n} \frac{D_{KL}(p_a^i||p_b^i)+D_{KL}(p_b^i||p_a^i)}{2}/n$, where p_a^i and p_b^i are the source distributions of two segments

[6] D_{2024}: 0.177, $D_{equitoken}$: 0.096, $D_{stratified}$: 0.197

[7] As documents may also be visited multiple times within an epoch, we use tau-b (Kendall, 1945) to account for ties. We truncate the longer of the two curricula where necessary.

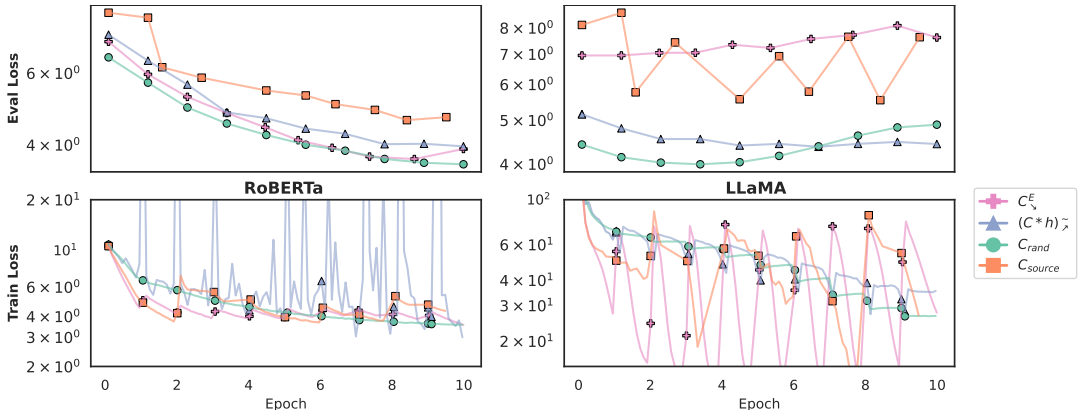


Figure 6: Train- and evaluation loss of baselines and influence curricula for $D_{stratified}$.

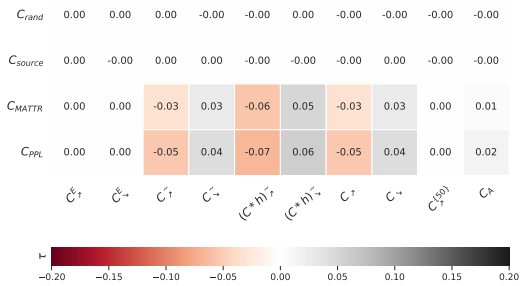


Figure 7: Rank-similarity between influence-curricula and baselines: Mean Kendall τ_b.

with a log-norm filter before sorting ($(C * h)^{\sim}_{\searrow}$) has a marginal positive, but insignificant effect on the similarity to baselines (+0.016 w.r.t. $C^{\sim}_{\searrow}$ for C_{MATTR}, +0.013 w.r.t C_{PPL}).

5 Discussion

Our results indicate that curricula based on training data influence estimates can be viable from a performance perspective; however, they are only so if paired with non-developmentally plausible coverage strategies (i.e., ones roughly inspired by how humans acquire language), in which the full training data is visited once per epoch: When specifically comparing the handcrafted- (C_{source}), and the two synthetic source-difficulty curricula ($C^{E}_{\nearrow}$, $C^{E}_{\searrow}$), it is evident that our sorting strategy based on training dynamics was unable to compensate for this less effective human-centered form of scheduling in terms of performance. Future work should therefore explore coverage strategies that more effectively balance model performance and developmentally plausible scheduling.

The observation that the ascending versions of the same strategy do not consistently outperform the descending versions (e.g., $C_{\searrow}$ and $C_{\nearrow}$) and vice versa suggests that the observed increase in performance might not stem from the specific sorting order (by increasing or decreasing influence), but rather from an **improved grouping of examples:** examples of similar influence are more likely located in the same batch. This would also explain the competitive performance of sorting by the model-agnostic difficulty heuristic C_{MATTR} on D_{2024} and $D_{stratified}$.

6 Conclusion

In this work, we study curriculum learning for language model pretraining and propose a novel type of curricula based on training data influence, which **outperforms training in random order** by up to 12.42 pp for RoBERTa models ($C_{\{50\}}$, D_{2024}) and up to 4.62 pp for Llama models ($C_{\nearrow}$, $D_{stratified}$). In contrast to recent experiments with handcrafted curricula, our results indicate that curriculum learning with our method has **potential to improve data efficiency in low-resource settings**.

Through an analysis of the data distribution in our curricula derived from influence estimates, we find that **their source composition does not strongly vary over time**, contrasting that of existing source-difficulty curricula, which are typically designed to decrease the proportion of child-directed speech in later epochs (replacing it with more complex text). Furthermore, by conducting an analysis of training- and evaluation loss trajectories, we have observed that the severe spikes in training loss seen with this form of curriculum learning are not significantly correlated with model performance on downstream benchmarks. Lastly, we explore how the ordering of examples with influence curricula correlates with existing sorting heuristics, finding that our measure is **inversely**

correlated to example difficulty (i.e., higher influence implies lower difficulty). In conclusion, our results suggest that curricula based on training data influence estimates can be viable from a performance perspective, but, their success may be attributed to training dynamics rather than increased developmental plausibility.

Limitations

We use a two-step approach to estimating training data influence: we first pre-train a model in random order, and subsequently extract the loss-gradients we utilize for influence estimation (one example at a time). We opted for this implementation to simplify our experimental setup, as our primary focus was on studying curriculum learning rather than minimizing training time. To improve computational efficiency within our framework, one could reuse (mini-batch) gradients from model training for influence estimation. We provide additional details on runtime in Appendix A.

In Section 4.2, where we study the data mix of our curricula, we observe that our influence curricula are highly sensitive to the source distribution of the dataset. If future work has an intention to use a similar influence estimation method for data cleaning or selection (as we did in $C^{\{50\}}$), it should explore measures to ensure appropriate data balancing. In our setup, the failure to do so primarily results in lower benchmark performance for $C^{\{50\}}$.

Lastly, our experiments are based on relatively small language models and datasets due to the lack of large-scale pre-training datasets that both cover and categorize examples across different difficulty levels. However, with D_{2024} we include a dataset that is widely used and studied through the BabyLM challenge (see Charpentier et al., 2025).

Acknowledgments

The present research was funded by the Go!Digital 3.0 grant program of the Austrian Academy of Sciences (GD3.0_2021-18_CogML) and by the Vienna Science and Technology Fund (WWTF)[10.47379/VRG19008] "Knowledge-infused Deep Learning for Natural Language Processing".

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA Corpus: Building Parallel Language Resources for the Educational Domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. 2021. Deep Learning Through the Lens of Example Difficulty. In *Advances in Neural Information Processing Systems*, volume 34, pages 10876–10889. Curran Associates, Inc.

Irina Bejan, Artem Sokolov, and Katja Filippova. 2023. Make Every Example Count: On the Stability and Utility of Self-Influence for Learning from Noisy NLP Datasets. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10107–10121, Singapore. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA. Association for Computing Machinery.

Eden Bensaid, Mauro Martino, Benjamin Hoover, and Hendrik Strobelt. 2021. FairyTailor: A Multimodal Generative Framework for Storytelling. *arXiv preprint*. ArXiv:2108.04324 [cs].

Nasim Borazjanizadeh. 2023. Optimizing GPT-2 Pretraining on BabyLM Corpus with Difficulty-based Sentence Reordering. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 356–365, Singapore. Association for Computational Linguistics.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM Turns 3: Call for papers for the 2025 BabyLM workshop. *arXiv preprint*. ArXiv:2502.10645 [cs].

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [Call for Papers] The 2nd BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint*. ArXiv:2404.06214.

B. N. C. Consortium. 2007. British National Corpus, XML edition. Accepted: 2018-07-27 Artwork Medium: Digital bitstream Interview Medium: Digital bitstream Publisher: University of Oxford.

Michael A. Covington, , and Joe D. McFall. 2010. Cutting the Gordian Knot: The Moving-Average Type–Token Ratio (MATTR). *Journal of Quantitative Linguistics*, 17(2):94–100. Publisher: Routledge _eprint: https://doi.org/10.1080/09296171003643098.

Jeffrey L. Elman. 1993. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99.

Martin Gerlach and Francesc Font-Clos. 2018. A standardized Project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *arXiv preprint*. ArXiv:1812.08092 [cs].

Mohammad Amin Ghanizadeh and Mohammad Javad Dousti. 2024. Towards Data-Efficient Language Models: A Child-Inspired Approach to Language Learning. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 22–27, Miami, FL, USA. Association for Computational Linguistics.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, D. Kimbrough Oller, John H. L. Hansen, and Terrance D. Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

Guy Hacohen and Daphna Weinshall. 2019. On The Power of Curriculum Learning in Training Deep Networks. *arXiv preprint*. ArXiv:1904.03626 [cs].

Zayd Hammoudeh and Daniel Lowd. 2022. Identifying a Training-Set Attack's Target Using Renormalized Influence Estimation. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, pages 1367–1381, New York, NY, USA. Association for Computing Machinery.

Zayd Hammoudeh and Daniel Lowd. 2024. Training data influence analysis and estimation: a survey. *Machine Learning*, 113(5):2351–2403.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv preprint*. ArXiv:1511.02301 [cs].

Valentin Hofmann, Leonie Weissweiler, David Mortensen, Hinrich Schütze, and Janet Pierrehumbert. 2024. Derivational Morphology Reveals Analogical Generalization in Large Language Models. *arXiv preprint*. ArXiv:2411.07990 [cs].

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the Second BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. *arXiv preprint*. ArXiv:2412.05149 [cs] version: 1.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning More Grammar With Small-Scale Child-Directed Language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of World Knowledge (EWOK): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint*. ArXiv:2405.09605 [cs] version: 1.

Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C. Mozer. 2021. Characterizing Structural Regularities of Labeled Data in Overparameterized Models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5034–5044. PMLR. ISSN: 2640-3498.

M. G. Kendall. 1945. THE TREATMENT OF TIES IN RANKING PROBLEMS. *Biometrika*, 33(3):239–251.

Najoung Kim and Sebastian Schuster. 2023. Entity Tracking in Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-Paced Learning for Latent Variable Models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Conglong Li, Minjia Zhang, and Yuxiong He. 2022. The Stability-Efficiency Dilemma: Investigating Sequence Length Warmup for Training GPT Models. *Advances in Neural Information Processing Systems*, 35:26736–26750.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint*. ArXiv:1907.11692 [cs].

Brian MacWhinney. 2014. *The Childes Project*, 0 edition. Psychology Press.

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – Curriculum Learning for Infant-inspired Model Building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 112–127, Singapore. Association for Computational Linguistics.

Maggie Mi. 2023. Mmi01 at The BabyLM Challenge: Linguistically Motivated Curriculum Learning for Pretraining in Low-Resource Settings. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 269–278, Singapore. Association for Computational Linguistics.

Miyu Oba, Akari Haga, Akiyo Fukatsu, and Yohei Oseki. 2023. BabyLM Challenge: Curriculum learning based on sentence complexity approximating language acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 290–297, Singapore. Association for Computational Linguistics.

Mattia Opper, J. Morrison, and N. Siddharth. 2023. On the effect of curriculum learning with developmental data for grammar acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 346–355, Singapore. Association for Computational Linguistics.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. TRAK: Attributing Model Behavior at Scale. In *Proceedings of the 40th International Conference on Machine Learning*, pages 27074–27113. PMLR. ISSN: 2640-3498.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based Curriculum Learning for Neural Machine Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating Training Data Influence by Tracing Gradient Descent. In *Advances in Neural Information Processing Systems*, volume 33, pages 19920–19930. Curran Associates, Inc.

Anastasiia Sedova, Lena Zellinger, and Benjamin Roth. 2023. Learning with Noisy Labels by Adaptive Gradient-Based Outlier Removal. In *Machine Learning and Knowledge Discovery in Databases: Research Track*, pages 237–253. Springer, Cham. ISSN: 1611-3349.

Valentin I. Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374. Place: Cambridge, MA Publisher: MIT Press.

Lukas Thoma, Ivonne Weyers, Erion Çano, Stefan Schweter, Jutta L Mueller, and Benjamin Roth. 2023. CogMemLM: Human-Like Memory Mechanisms Improve Performance and Cognitive Plausibility of LLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 180–185, Singapore. Association for Computational Linguistics.

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby Llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint*. ArXiv:2302.13971 [cs].

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *arXiv preprint*. ArXiv:1804.07461.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for Papers – The BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint*. ArXiv:2301.11796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023b. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2020. When Do Curricula Work? In *International Conference on Learning Representations*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting Influential Data for Targeted Instruction Tuning. *arXiv preprint*. ArXiv:2402.04333 [cs].

Chih-Kuan Yeh, Ankur Taly, Mukund Sundararajan, Frederick Liu, and Pradeep Ravikumar. 2022. First is Better Than Last for Language Data Influence. *arXiv preprint*. ArXiv:2202.11844 [cs].

A Implementation Details

Influence Estimation

To enable influence estimation for the RoBERTa models, which are trained with dynamic masking (tokens are masked differently at each epoch), we implement a custom *Data Collator* for use with the Hugging Face *Trainer*: This collator makes masking reproducible by computing a hash based on the document and the epoch number.

Runtime

Pre-training of all 84 models took 195 hours on 4 NVIDIA H100 GPUs (approximately 2h20m per model). The runtime of the influence estimation step, which is only required once per dataset, depends on the number of documents. On average (across model architectures and datasets) it amounts to roughly 44.3h if estimation is run sequentially for each checkpoint, or just under 5h if run in parallel. Sequential runtime would amount to 7h45h for $D_{equitoken}$, 109h for D_{2024} (both ran on NVIDIA H100 GPUs), and 149h30min on $D_{stratified}$ (ran on a lower-spec NVIDIA V100 GPUs), totaling 266 GPU hours overall.

	RoBERTa	**LLaMA**
Vocabulary size	52k	
Hidden size	768	
Number of layers	12	
Number of attention heads	12	
Initializer range	0.02	
Tie word embeddings	True	
Max position embeddings	514	256
Intermediate (FFN) size	3072	2048
Norm epsilon	1e-5	1e-6
Attention dropout	0.1	0
Activation function	`gelu`	`silu`
Hidden dropout	0.1	-
FP16	False	
Per Device Batch Size	32	
Gradient Accumulation Steps	16	
GPUs	4	
Adam β_1	0.9	
Adam β_2	0.98	
Adam ϵ	1e-6	
Weight Decay ϵ	0.01	
Learning rate	5e-4	7e-4
Scheduler	polynomial	cosine

Table 3: Training parameters used for all models.

B Loss Trajectories

Our curricula sort examples based on their influence, which may inadvertently reduce example diversity within training batches. We hypothesize that this led to the substantial training loss spikes observed. While one can measure loss during training with a separate evaluation set (as we have done), this adds significant overhead during training. To analyze whether training loss spikes are still indicative of training instability for curriculum learning, i.e. wether their severity ultimately impacts benchmark performance, we employ the *loss ratio* metric proposed by Li et al. (2022), as a measure of training instability, which compares the loss at the current step s to the lowest loss achieved in any prior step: $lr(s) = \frac{\ell(s)}{\min_{s'<s} \ell(s')}$. Intuitively (if training in random order), one would expect models with high loss ratios to have lower benchmark performance. However, an analysis of the corelation between a curriculum's gain in benchmark performance (over training in random order) and this loss-ratio indeed does not reveal a significant negative Spearman rank correlation for any dataset: D_{2024}: 0.177; $D_{equitoken}$: 0.096; $D_{stratified}$: 0.197.

C Complementary Figures

This section presents complementary figures for RoBERTa or Llama models, with the respective other model type included in the main body of our paper.

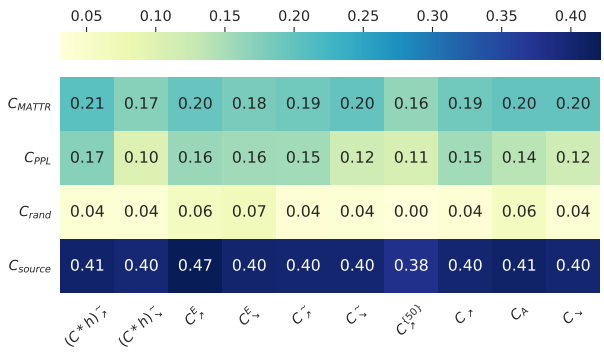


Figure 8: Comparison of curriculum stage distributions: Average Jensen–Shannon divergence between 1000 segments of two given curricula for RoBERTa models. Lower values indicate more similar stage distributions.

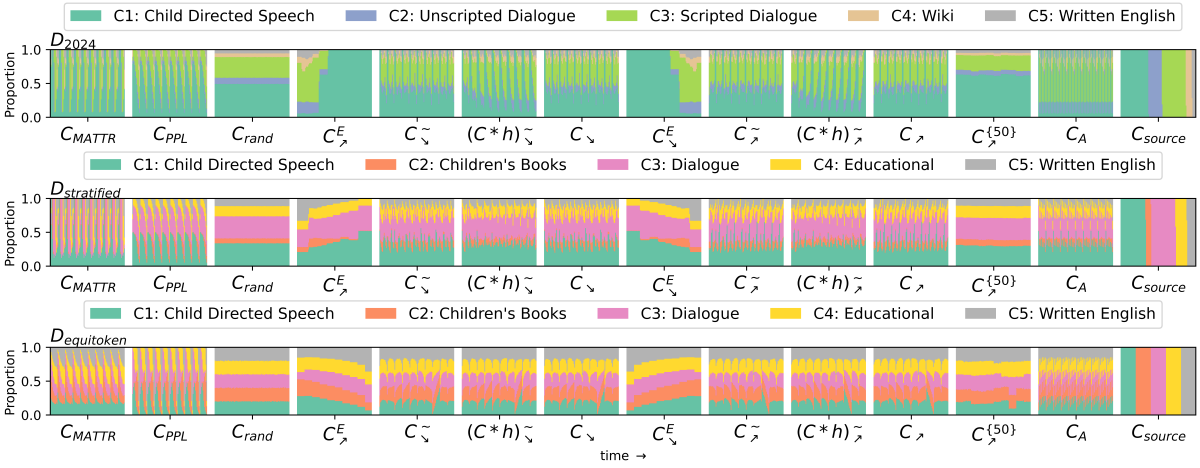


Figure 9: Dataset mix of curricula for RoBERTa models. We trace back documents to the stages defined in Table 1.

D Full Benchmark Results and Loss Trajectories

Table 4: Macro-average gain in accuracy over the corresponding random curriculum.

Curriculum	Dataset	Architecture	Improvement	p-val	Model acc	Random acc
C_{rand}	D_{2024}	RoBERTa	+0.00 pp	-	0.466	-
C_{rand}	$D_{equitoken}$	RoBERTa	+0.00 pp	-	0.492	-
C_{rand}	$D_{stratified}$	RoBERTa	+0.00 pp	-	0.512	-
C_{rand}	$D_{equitoken}$	Llama	+0.00 pp	-	0.523	-
C_{rand}	$D_{stratified}$	Llama	+0.00 pp	-	0.536	-
C_{rand}	D_{2024}	Llama	+0.00 pp	-	0.541	-
$C^{E}_{\nearrow}$	$D_{equitoken}$	Llama	-5.02 pp**	0.033	0.473	0.523
$C^{E}_{\nearrow}$	D_{2024}	Llama	-4.84 pp***	0.004	0.493	0.541
$C^{E}_{\nearrow}$	$D_{stratified}$	Llama	-4.79 pp***	0.005	0.488	0.536
$C^{E}_{\nearrow}$	D_{2024}	Llama	-3.83 pp*	0.065	0.503	0.541
$C^{E}_{\searrow}$	$D_{stratified}$	Llama	-3.11 pp***	0.002	0.504	0.536
$C^{E}_{\searrow}$	$D_{equitoken}$	Llama	-3.10 pp	0.100	0.492	0.523
$(C*h)^{\sim}_{\searrow}$	$D_{equitoken}$	Llama	-1.82 pp	0.400	0.505	0.523

Continued on next page

Continued from previous page

Curriculum	Dataset	Architecture	Improvement	p-val	Model acc	Random acc
C_{source}	$D_{stratified}$	Llama	-1.39 pp	0.167	0.522	0.536
$C^{\{50\}}_{\nearrow}$	$D_{equitoken}$	Llama	-1.24 pp	0.504	0.511	0.523
C_{MATTR}	$D_{equitoken}$	Llama	-0.72 pp	0.293	0.516	0.523
C_{PPL}	$D_{equitoken}$	Llama	-0.65 pp	0.431	0.517	0.523
C_{A}	$D_{equitoken}$	RoBERTa	-0.55 pp	0.726	0.487	0.492
C_{PPL}	$D_{stratified}$	RoBERTa	-0.28 pp	0.877	0.510	0.512
C_{source}	$D_{equitoken}$	Llama	-0.12 pp	0.856	0.522	0.523
$C_{\searrow}$	D_{2024}	Llama	-0.02 pp	0.991	0.541	0.541
C_{A}	D_{2024}	Llama	+0.17 pp	0.796	0.543	0.541
$C^{\{50\}}_{\nearrow}$	D_{2024}	Llama	+0.21 pp	0.918	0.543	0.541
$C^{E}_{\nearrow}$	$D_{stratified}$	RoBERTa	+0.36 pp	0.801	0.516	0.512
$(C*h)^{\sim}_{\nearrow}$	$D_{equitoken}$	Llama	+0.42 pp	0.848	0.527	0.523
C_{A}	$D_{equitoken}$	Llama	+0.53 pp	0.813	0.528	0.523
C_{PPL}	D_{2024}	Llama	+0.72 pp	0.317	0.548	0.541
$C^{\{50\}}_{\nearrow}$	$D_{stratified}$	Llama	+0.92 pp	0.619	0.545	0.536
C_{source}	D_{2024}	Llama	+1.07 pp	0.242	0.552	0.541
$(C*h)^{\sim}_{\nearrow}$	D_{2024}	Llama	+1.29 pp	0.504	0.554	0.541
$C^{\sim}_{\searrow}$	$D_{equitoken}$	Llama	+1.31 pp	0.150	0.536	0.523
$C^{\sim}_{\searrow}$	D_{2024}	Llama	+1.37 pp	0.477	0.555	0.541
$C^{\sim}_{\nearrow}$	$D_{equitoken}$	Llama	+1.50 pp	0.494	0.538	0.523
$C^{\sim}_{\nearrow}$	$D_{stratified}$	Llama	+1.73 pp***	0.007	0.553	0.536
$C^{\sim}_{\nearrow}$	D_{2024}	Llama	+1.77 pp	0.362	0.559	0.541
$C_{\nearrow}$	D_{2024}	Llama	+1.78 pp	0.371	0.559	0.541
C_{MATTR}	$D_{stratified}$	Llama	+1.86 pp**	0.029	0.554	0.536
$C^{E}_{\nearrow}$	$D_{equitoken}$	RoBERTa	+1.93 pp	0.236	0.512	0.492
$C_{\searrow}$	$D_{equitoken}$	Llama	+2.29 pp***	0.006	0.546	0.523
$C_{\searrow}$	$D_{stratified}$	Llama	+2.37 pp***	0.002	0.559	0.536
$(C*h)^{\sim}_{\searrow}$	$D_{stratified}$	Llama	+2.41 pp***	0.001	0.560	0.536
C_{MATTR}	$D_{equitoken}$	RoBERTa	+2.62 pp	0.138	0.518	0.492
$C^{E}_{\nearrow}$	D_{2024}	RoBERTa	+3.02 pp	0.124	0.496	0.466
C_{MATTR}	D_{2024}	Llama	+3.07 pp***	0.000	0.572	0.541
$(C*h)^{\sim}_{\nearrow}$	$D_{stratified}$	Llama	+3.08 pp	0.122	0.566	0.536
$(C*h)^{\sim}_{\nearrow}$	$D_{equitoken}$	RoBERTa	+3.10 pp	0.123	0.523	0.492
$C^{\sim}_{\searrow}$	$D_{equitoken}$	RoBERTa	+3.12 pp*	0.079	0.523	0.492
$C_{\nearrow}$	$D_{equitoken}$	Llama	+3.16 pp	0.142	0.555	0.523
$C^{\sim}_{\nearrow}$	$D_{equitoken}$	RoBERTa	+3.31 pp*	0.077	0.525	0.492
$C^{\sim}_{\nearrow}$	$D_{stratified}$	RoBERTa	+3.32 pp	0.166	0.546	0.512
$C_{\nearrow}$	$D_{stratified}$	RoBERTa	+3.51 pp	0.140	0.548	0.512
$C_{\searrow}$	$D_{equitoken}$	RoBERTa	+3.57 pp**	0.050	0.528	0.492
C_{MATTR}	$D_{stratified}$	RoBERTa	+3.81 pp	0.126	0.551	0.512
C_{source}	$D_{stratified}$	RoBERTa	+3.89 pp	0.120	0.551	0.512
C_{PPL}	$D_{equitoken}$	RoBERTa	+3.92 pp**	0.032	0.531	0.492
C_{PPL}	$D_{stratified}$	Llama	+3.97 pp***	0.000	0.575	0.536
$(C*h)^{\sim}_{\nearrow}$	$D_{equitoken}$	RoBERTa	+4.00 pp*	0.050	0.532	0.492
C_{source}	$D_{equitoken}$	RoBERTa	+4.12 pp*	0.052	0.533	0.492
$C^{E}_{\searrow}$	$D_{equitoken}$	RoBERTa	+4.16 pp**	0.041	0.534	0.492
$C_{\searrow}$	$D_{stratified}$	RoBERTa	+4.16 pp*	0.079	0.554	0.512
C_{A}	$D_{stratified}$	Llama	+4.18 pp***	0.000	0.577	0.536
$C^{\sim}_{\searrow}$	$D_{stratified}$	Llama	+4.18 pp***	0.000	0.577	0.536
$C^{\sim}_{\searrow}$	$D_{stratified}$	RoBERTa	+4.26 pp*	0.094	0.555	0.512
$(C*h)^{\sim}_{\searrow}$	D_{2024}	Llama	+4.34 pp**	0.028	0.584	0.541
$C^{\{50\}}_{\nearrow}$	$D_{equitoken}$	RoBERTa	+4.36 pp**	0.039	0.536	0.492
$C^{E}_{\searrow}$	$D_{stratified}$	RoBERTa	+4.40 pp*	0.052	0.556	0.512
$(C*h)^{\sim}_{\searrow}$	$D_{stratified}$	RoBERTa	+4.47 pp*	0.072	0.557	0.512
$C^{\{50\}}_{\nearrow}$	$D_{stratified}$	RoBERTa	+4.52 pp*	0.067	0.558	0.512
$C_{\nearrow}$	$D_{equitoken}$	RoBERTa	+4.54 pp**	0.031	0.538	0.492
$C_{\nearrow}$	$D_{stratified}$	Llama	+4.62 pp***	0.000	0.582	0.536
C_{A}	$D_{stratified}$	RoBERTa	+6.67 pp***	0.004	0.579	0.512
$(C*h)^{\sim}_{\nearrow}$	$D_{stratified}$	RoBERTa	+7.96 pp***	0.000	0.592	0.512
$C_{\searrow}$	D_{2024}	RoBERTa	+8.72 pp***	0.004	0.553	0.466
$(C*h)^{\sim}_{\nearrow}$	D_{2024}	RoBERTa	+8.74 pp***	0.002	0.553	0.466
$C^{\sim}_{\searrow}$	D_{2024}	RoBERTa	+9.13 pp***	0.002	0.557	0.466
$C^{\sim}_{\nearrow}$	D_{2024}	RoBERTa	+9.36 pp***	0.000	0.559	0.466

Continued on next page

Continued from previous page

Curriculum	Dataset	Architecture	Improvement	p-val	Model acc	Random acc
C_{PPL}	D_{2024}	RoBERTa	+9.49 pp***	0.000	0.561	0.466
C_{A}	D_{2024}	RoBERTa	+10.19 pp***	0.001	0.568	0.466
$(C * h)^{\sim}_{\searrow}$	D_{2024}	RoBERTa	+10.71 pp***	0.000	0.573	0.466
C_{MATTR}	D_{2024}	RoBERTa	+10.97 pp***	0.000	0.575	0.466
$C^{E}_{\searrow}$	D_{2024}	RoBERTa	+10.98 pp***	0.000	0.576	0.466
$C_{\nearrow}$	D_{2024}	RoBERTa	+11.00 pp***	0.000	0.576	0.466
C_{source}	D_{2024}	RoBERTa	+11.77 pp***	0.000	0.583	0.466
$C^{\{50\}}_{\nearrow}$	D_{2024}	RoBERTa	+12.42 pp***	0.000	0.590	0.466
E_{mixed}	ext	gpt-bert	-	-	0.498	-
E_{causal}	ext	gpt-bert	-	-	0.502	-
E_{masked}	ext	gpt-bert	-	-	0.504	-
E_{gpt2}	ext	-	-	-	0.551	-

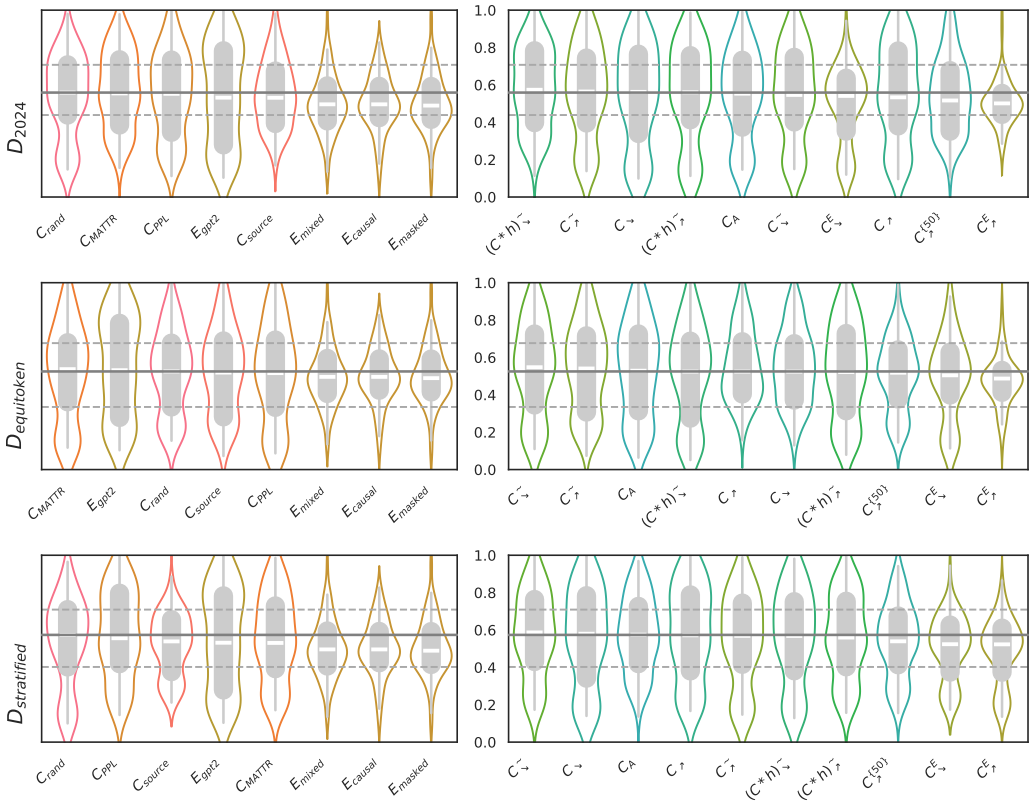


Figure 10: Benchmark results for Llama models.

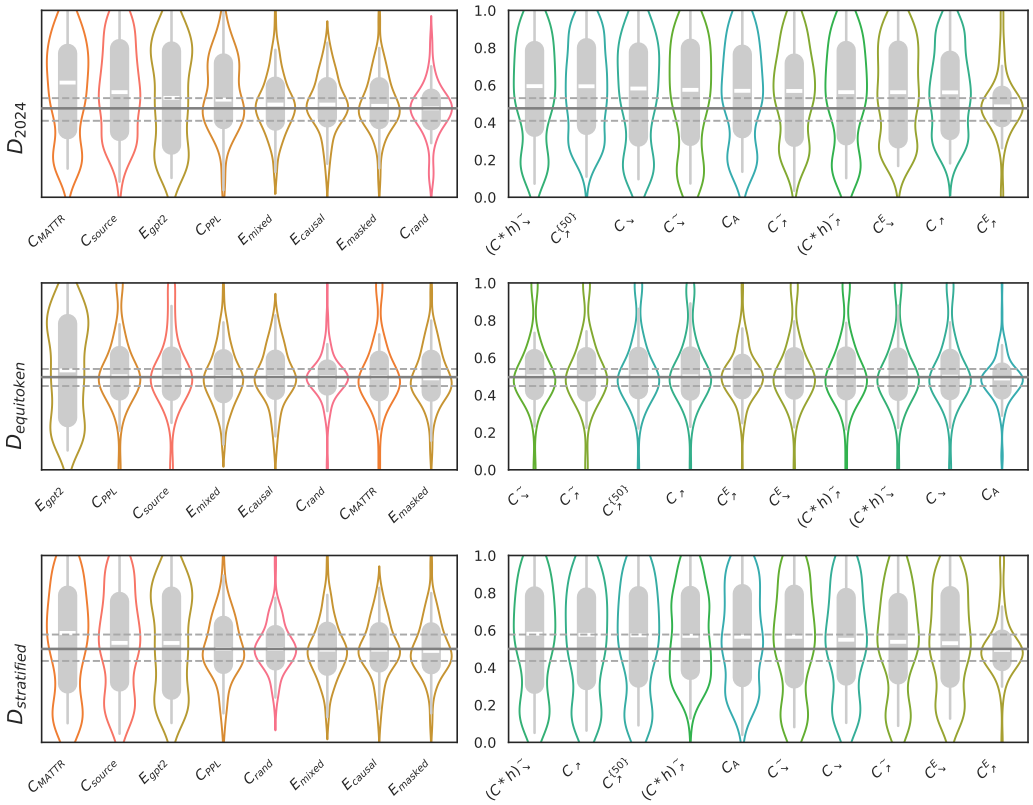


Figure 11: Benchmark results for RoBERTa models.

Table 5: Macro accuracy for Llama models across tasks, per benchmark and overall. $E.$ denotes baseline models from the BabyLM challange, the fine-tuning evaluation pipeline fails for the E_{gpt2} model.

Curriculum	Dataset	(Super) GLUE	blimp_filtered	supplement_filtered	entity_tracking	ewok_filtered	wug_adj_nominalization	Macro acc
$(C*h)^{\sim}_{\searrow}$	D_{2024}	0.579	0.688	0.559	0.302	0.509	0.570	**0.584**
$C_{\nearrow}$	$D_{stratified}$	0.573	**0.715**	0.546	0.208	0.503	0.600	0.582
C_{A}	$D_{stratified}$	0.575	0.675	0.575	0.306	0.507	0.560	0.577
$C^{\sim}_{\searrow}$	$D_{stratified}$	0.573	0.695	0.558	0.242	**0.519**	0.495	0.577
C_{PPL}	$D_{stratified}$	0.573	0.696	0.532	0.239	0.510	0.480	0.575
C_{MATTR}	D_{2024}	0.573	0.671	0.551	0.295	0.507	0.550	0.572
$(C*h)^{\sim}_{\nearrow}$	$D_{stratified}$	0.572	0.678	0.567	0.245	0.501	0.540	0.566
$(C*h)^{\sim}_{\searrow}$	$D_{stratified}$	0.573	0.691	0.542	0.169	0.507	0.510	0.560
$C_{\searrow}$	$D_{stratified}$	0.567	0.694	0.533	0.164	0.512	0.420	0.559
$C_{\nearrow}$	D_{2024}	0.575	0.686	0.566	0.184	0.494	0.500	0.559
$C^{\sim}_{\nearrow}$	D_{2024}	0.571	0.683	0.566	0.184	0.506	0.565	0.559
$C^{\sim}_{\searrow}$	D_{2024}	0.571	0.679	0.571	0.176	0.506	0.500	0.555
$C_{\nearrow}$	$D_{equitoken}$	0.575	0.618	0.514	0.389	0.489	0.555	0.555
C_{MATTR}	$D_{stratified}$	0.571	0.663	0.539	0.227	0.516	0.495	0.554
$(C*h)^{\sim}_{\nearrow}$	D_{2024}	0.570	0.692	0.536	0.136	0.511	0.515	0.554
$C^{\sim}_{\nearrow}$	$D_{stratified}$	0.568	0.684	0.514	0.169	0.499	0.535	0.553
C_{source}	D_{2024}	0.576	0.628	0.503	0.336	0.505	0.560	0.552
E_{gpt2}	ext	nan	0.673	**0.591**	0.189	0.498	0.390	0.551
C_{PPL}	D_{2024}	0.582	0.655	0.508	0.226	0.499	0.655	0.548
$C_{\searrow}$	$D_{equitoken}$	0.577	0.615	0.528	0.336	0.501	0.685	0.546
$C^{\{50\}}_{\nearrow}$	$D_{stratified}$	0.575	0.633	0.540	0.267	0.510	0.635	0.545
C_{A}	D_{2024}	0.573	0.660	0.520	0.178	0.506	0.635	0.543
$C^{\{50\}}_{\nearrow}$	D_{2024}	0.572	0.618	0.541	0.314	0.497	0.635	0.543
C_{rand}	D_{2024}	0.572	0.658	0.497	0.193	0.500	0.440	0.541
$C_{\searrow}$	D_{2024}	0.573	0.674	0.521	0.133	0.500	0.530	0.541
$C^{\sim}_{\nearrow}$	$D_{equitoken}$	0.577	0.634	0.561	0.234	0.503	0.465	0.538
C_{rand}	$D_{stratified}$	0.576	0.662	0.517	0.142	0.500	0.550	0.536
$C^{\sim}_{\searrow}$	$D_{equitoken}$	0.578	0.650	0.514	0.179	0.502	0.620	0.536
C_{A}	$D_{equitoken}$	0.577	0.634	0.547	0.184	0.492	0.625	0.528
$(C*h)^{\sim}_{\nearrow}$	$D_{equitoken}$	0.577	0.638	0.559	0.168	0.495	0.485	0.527
C_{rand}	$D_{equitoken}$	0.579	0.615	0.548	0.215	0.493	0.625	0.523
C_{source}	$D_{equitoken}$	0.577	0.609	0.480	0.244	0.499	0.615	0.522
C_{source}	$D_{stratified}$	0.577	0.593	0.479	0.286	0.518	0.570	0.522
C_{PPL}	$D_{equitoken}$	0.582	0.635	0.490	0.129	0.498	0.610	0.517
C_{MATTR}	$D_{equitoken}$	0.579	0.627	0.529	0.141	0.498	0.665	0.516
$C^{\{50\}}_{\nearrow}$	$D_{equitoken}$	0.579	0.592	0.535	0.228	0.503	0.495	0.511
$(C*h)^{\sim}_{\searrow}$	$D_{equitoken}$	0.578	0.610	0.542	0.136	0.502	0.520	0.505
E_{masked}	ext	**0.665**	0.508	0.483	**0.419**	0.502	**0.965**	0.504

Continued on next page

Continued from previous page

Curriculum	Dataset	(Super) GLUE	blimp_filtered	supplement_filtered	entity_tracking	ewok_filtered	wug_adj_nominalization	Macro acc
$C^{E}_{\searrow}$	$D_{stratified}$	0.576	0.577	0.535	0.242	0.500	0.555	0.504
$C^{E}_{\nearrow}$	D_{2024}	0.570	0.518	0.499	0.412	0.506	0.925	0.503
E_{causal}	ext	0.654	0.514	0.449	0.412	0.502	0.770	0.502
E_{mixed}	ext	0.660	0.505	0.459	0.414	0.500	0.780	0.498
$C^{E}_{\searrow}$	D_{2024}	0.577	0.586	0.486	0.152	0.512	0.635	0.493
$C^{E}_{\searrow}$	$D_{equitoken}$	0.576	0.587	0.507	0.148	0.502	0.590	0.492
$C^{E}_{\nearrow}$	$D_{stratified}$	0.573	0.562	0.510	0.201	0.505	0.640	0.488
$C^{E}_{\nearrow}$	$D_{equitoken}$	0.570	0.471	0.502	0.415	0.502	0.685	0.473

Table 6: Macro accuracy for RoBERTa models across tasks, per benchmark and overall. $E.$ denotes baseline models from the BabyLM challange, the fine-tuning evaluation pipeline fails for the E_{gpt2} model.

Curriculum	Dataset	(Super) GLUE	blimp_filtered	supplement_filtered	entity_tracking	ewok_filtered	wug_adj_nominalization	Macro acc
$(C*h)^{\sim}_{\nearrow}$	$D_{stratified}$	0.650	0.694	0.535	0.307	0.507	0.570	**0.592**
$C^{\{50\}}_{\nearrow}$	D_{2024}	0.634	0.700	0.578	0.268	0.501	0.690	0.590
C_{source}	D_{2024}	0.635	0.683	0.563	0.290	0.507	0.670	0.583
C_{A}	$D_{stratified}$	0.629	0.698	0.555	0.238	0.497	0.460	0.579
$C_{\nearrow}$	D_{2024}	0.643	0.664	0.535	0.309	0.504	0.780	0.576
$C^{E}_{\searrow}$	D_{2024}	0.641	0.689	0.530	0.237	0.500	0.655	0.576
C_{MATTR}	D_{2024}	0.642	0.701	0.556	0.186	0.505	0.715	0.575
$(C*h)^{\sim}_{\searrow}$	D_{2024}	0.645	**0.702**	0.570	0.171	0.500	0.680	0.573
C_{A}	D_{2024}	0.636	0.691	0.529	0.187	0.505	0.675	0.568
C_{PPL}	D_{2024}	0.633	0.614	0.534	0.402	0.500	0.620	0.561
$C^{\sim}_{\nearrow}$	D_{2024}	0.639	0.658	0.567	0.239	0.501	0.715	0.559
$C^{\{50\}}_{\nearrow}$	$D_{stratified}$	0.647	0.690	0.571	0.126	0.503	0.650	0.558
$C^{\sim}_{\nearrow}$	D_{2024}	0.637	0.692	0.561	0.117	0.502	0.775	0.557
$(C*h)^{\sim}_{\searrow}$	$D_{stratified}$	0.650	0.689	0.538	0.140	0.502	0.560	0.557
$C^{E}_{\nearrow}$	$D_{stratified}$	0.647	0.680	0.563	0.162	0.499	0.530	0.556
$C^{\sim}_{\searrow}$	$D_{stratified}$	0.638	0.690	0.556	0.128	0.503	0.460	0.555
$C_{\searrow}$	$D_{stratified}$	0.643	0.679	0.559	0.149	0.500	0.610	0.554
$C_{\searrow}$	D_{2024}	0.639	0.675	0.565	0.149	0.501	0.765	0.553
$(C*h)^{\sim}_{\nearrow}$	D_{2024}	0.626	0.675	0.571	0.148	**0.511**	0.745	0.553
C_{MATTR}	$D_{stratified}$	0.648	0.683	0.535	0.127	0.502	0.505	0.551
C_{source}	$D_{stratified}$	0.636	0.668	0.560	0.177	0.499	0.560	0.551
E_{gpt2}	ext	nan	0.673	**0.591**	0.189	0.498	0.390	0.551
$C_{\nearrow}$	$D_{stratified}$	0.644	0.677	0.530	0.125	0.499	0.715	0.548

Continued on next page

Curriculum	Dataset	(Super) GLUE	blimp_filtered	supplement_filtered	entity_tracking	ewok_filtered	wug_adj_nominalization	Macro acc
$C^{\sim}_{\nearrow}$	$D_{stratified}$	0.638	0.677	0.535	0.123	0.502	0.540	0.546
$C_{\nearrow}$	$D_{equitoken}$	0.611	0.575	0.476	0.409	0.501	0.925	0.538
$C^{\{50\}}_{\nearrow}$	$D_{equitoken}$	0.605	0.575	0.488	0.409	0.500	0.690	0.536
$C^{E}_{\searrow}$	$D_{equitoken}$	0.600	0.573	0.485	0.406	0.502	0.720	0.534
C_{source}	$D_{equitoken}$	0.609	0.569	0.488	0.407	0.500	0.855	0.533
$(C * h)^{\sim}_{\nearrow}$	$D_{equitoken}$	0.612	0.570	0.471	0.409	0.498	0.690	0.532
C_{PPL}	$D_{equitoken}$	0.605	0.566	0.486	0.411	0.501	0.770	0.531
$C_{\searrow}$	$D_{equitoken}$	0.606	0.564	0.475	0.409	0.496	0.690	0.528
$C^{\sim}_{\nearrow}$	$D_{equitoken}$	0.602	0.558	0.484	0.411	0.500	0.720	0.525
$C^{\sim}_{\searrow}$	$D_{equitoken}$	0.612	0.559	0.450	0.409	0.491	0.600	0.523
$(C * h)^{\sim}_{\searrow}$	$D_{equitoken}$	0.614	0.557	0.484	0.406	0.501	0.495	0.523
C_{MATTR}	$D_{equitoken}$	0.603	0.552	0.455	0.409	0.499	0.525	0.518
$C^{E}_{\nearrow}$	$D_{stratified}$	0.594	0.550	0.462	0.414	0.492	0.400	0.516
C_{rand}	$D_{stratified}$	0.591	0.542	0.467	0.408	0.504	0.520	0.512
$C^{E}_{\nearrow}$	$D_{equitoken}$	0.605	0.531	0.506	0.411	0.501	0.795	0.512
C_{PPL}	$D_{stratified}$	0.645	0.535	0.488	0.411	0.497	0.195	0.510
E_{masked}	ext	**0.665**	0.508	0.483	**0.419**	0.502	**0.965**	0.504
E_{causal}	ext	0.654	0.514	0.449	0.412	0.502	0.770	0.502
E_{mixed}	ext	0.660	0.505	0.459	0.414	0.500	0.780	0.498
$C^{E}_{\nearrow}$	D_{2024}	0.591	0.516	0.459	0.407	0.497	0.540	0.496
C_{rand}	$D_{equitoken}$	0.595	0.506	0.438	0.418	0.494	0.660	0.492
C_{A}	$D_{equitoken}$	0.597	0.501	0.441	0.413	0.489	0.510	0.487
C_{rand}	D_{2024}	0.601	0.462	0.465	0.409	0.507	0.490	0.466

Table 7: Average % R^2 gain for Llama models in the reading benchmarks (not included in the main paper). $E.$ denotes baseline models from the BabyLM challenge.

Curriculum	Dataset	Eye Tracking Score	Self-Paced Reading Score	Avg
E_{causal}	ext	0.102	**0.029**	**0.065**
E_{masked}	ext	**0.103**	0.027	0.065
E_{mixed}	ext	0.099	0.025	0.062
$C^{E}_{\nearrow}$	$D_{equitoken}$	0.024	0.009	0.016
$C^{E}_{\nearrow}$	D_{2024}	0.021	0.010	0.016
C_{source}	$D_{stratified}$	0.011	0.001	0.006
$C^{E}_{\searrow}$	$D_{stratified}$	0.012	0.000	0.006
$C^{E}_{\searrow}$	D_{2024}	0.009	0.001	0.005
C_{source}	D_{2024}	0.006	0.001	0.003
$C^{\{50\}}_{\nearrow}$	$D_{equitoken}$	0.006	0.000	0.003
C_{rand}	$D_{equitoken}$	0.005	0.001	0.003
C_{source}	$D_{equitoken}$	0.005	0.001	0.003
C_{rand}	D_{2024}	0.005	0.001	0.003
$C_{\searrow}$	$D_{stratified}$	0.006	0.001	0.003
$C_{\searrow}$	$D_{equitoken}$	0.005	0.000	0.003
$C_{\searrow}$	D_{2024}	0.005	0.000	0.003
C_{PPL}	$D_{equitoken}$	0.006	0.001	0.003
C_{MATTR}	$D_{equitoken}$	0.005	0.000	0.003
$C^{\{50\}}_{\nearrow}$	$D_{stratified}$	0.005	0.001	0.003
$(C*h)^{\sim}_{\nearrow}$	D_{2024}	0.003	0.002	0.003
$C_{\nearrow}$	$D_{stratified}$	0.005	0.000	0.003
$(C*h)^{\sim}_{\nearrow}$	$D_{stratified}$	0.005	0.001	0.003
$(C*h)^{\sim}_{\searrow}$	$D_{equitoken}$	0.007	0.000	0.003
$C^{\sim}_{\searrow}$	$D_{equitoken}$	0.006	0.000	0.003
C_{PPL}	$D_{stratified}$	0.003	0.000	0.002
$(C*h)^{\sim}_{\searrow}$	D_{2024}	0.004	0.001	0.002
$(C*h)^{\sim}_{\searrow}$	$D_{stratified}$	0.005	0.000	0.002
C_{rand}	$D_{stratified}$	0.004	0.000	0.002
$C^{E}_{\searrow}$	$D_{equitoken}$	0.004	0.000	0.002
$C^{\sim}_{\nearrow}$	D_{2024}	0.003	0.000	0.002
$C^{\sim}_{\nearrow}$	$D_{equitoken}$	0.004	0.000	0.002
$C^{\sim}_{\nearrow}$	$D_{stratified}$	0.005	0.000	0.002
$C^{\sim}_{\searrow}$	D_{2024}	0.004	0.000	0.002
C_{PPL}	D_{2024}	0.003	0.001	0.002
$C^{\sim}_{\searrow}$	$D_{stratified}$	0.004	0.000	0.002
C_{A}	$D_{stratified}$	0.003	0.001	0.002
C_{A}	$D_{equitoken}$	0.004	0.000	0.002
C_{A}	D_{2024}	0.005	0.000	0.002
$(C*h)^{\sim}_{\nearrow}$	$D_{equitoken}$	0.005	0.000	0.002
$C_{\nearrow}$	$D_{equitoken}$	0.004	0.000	0.002
$C_{\nearrow}$	D_{2024}	0.002	0.000	0.001
C_{MATTR}	$D_{stratified}$	0.002	0.000	0.001
C_{MATTR}	D_{2024}	0.003	0.000	0.001
$C^{E}_{\nearrow}$	$D_{stratified}$	0.002	0.000	0.001
E_{gpt2}	ext	0.001	0.000	0.001
$C^{\{50\}}_{\nearrow}$	D_{2024}	0.001	0.002	0.001

Table 8: Average % R^2 gain for RoBERTa models in the reading benchmarks (not included in the main paper). $E.$ denotes baseline models from the BabyLM challenge.

Curriculum	Dataset	Eye Tracking Score	Self-Paced Reading Score	Avg
E_{causal}	ext	0.102	**0.029**	**0.065**
E_{masked}	ext	**0.103**	0.027	0.065
E_{mixed}	ext	0.099	0.025	0.062
$C^{E}_{\nearrow}$	$D_{stratified}$	0.076	0.015	0.046
$C^{E}_{\nearrow}$	D_{2024}	0.074	0.014	0.044
C_{rand}	$D_{stratified}$	0.070	0.016	0.043
$C_{\nearrow}$	$D_{stratified}$	0.075	0.009	0.042
C_{PPL}	$D_{stratified}$	0.071	0.012	0.041

Continued on next page

Continued from previous page

Curriculum	Dataset	Eye Tracking Score	Self-Paced Reading Score	Avg
C_{rand}	D_{2024}	0.064	0.011	0.037
C_{PPL}	D_{2024}	0.060	0.007	0.033
$C^{\sim}_{\nearrow}$	$D_{stratified}$	0.051	0.007	0.029
$C^{E}_{\nearrow}$	$D_{equitoken}$	0.045	0.011	0.028
$C_{\nearrow}$	D_{2024}	0.050	0.006	0.028
C_{A}	$D_{equitoken}$	0.045	0.012	0.028
C_{rand}	$D_{equitoken}$	0.041	0.012	0.027
$(C*h)^{\sim}_{\searrow}$	$D_{stratified}$	0.046	0.005	0.026
$C^{E}_{\searrow}$	$D_{stratified}$	0.043	0.004	0.024
$C^{E}_{\searrow}$	D_{2024}	0.045	0.003	0.024
$C^{\sim}_{\searrow}$	D_{2024}	0.039	0.007	0.023
$(C*h)^{\sim}_{\searrow}$	D_{2024}	0.039	0.005	0.022
C_{source}	D_{2024}	0.039	0.003	0.021
$C^{\sim}_{\nearrow}$	D_{2024}	0.035	0.007	0.021
C_{A}	D_{2024}	0.036	0.005	0.021
$C_{\searrow}$	$D_{stratified}$	0.036	0.003	0.020
$C^{\{50\}}_{\nearrow}$	$D_{stratified}$	0.034	0.004	0.019
C_{A}	$D_{stratified}$	0.034	0.003	0.018
$C^{\sim}_{\searrow}$	$D_{stratified}$	0.033	0.003	0.018
$C_{\searrow}$	D_{2024}	0.030	0.005	0.017
$C^{\{50\}}_{\nearrow}$	D_{2024}	0.033	0.002	0.017
$(C*h)^{\sim}_{\nearrow}$	$D_{stratified}$	0.031	0.002	0.016
C_{MATTR}	D_{2024}	0.029	0.003	0.016
C_{source}	$D_{stratified}$	0.024	0.003	0.014
$(C*h)^{\sim}_{\nearrow}$	D_{2024}	0.019	0.001	0.010
$C_{\searrow}$	$D_{equitoken}$	0.015	0.003	0.009
$(C*h)^{\sim}_{\searrow}$	$D_{equitoken}$	0.015	0.003	0.009
$C^{\sim}_{\nearrow}$	$D_{equitoken}$	0.015	0.003	0.009
C_{source}	$D_{equitoken}$	0.016	0.003	0.009
C_{MATTR}	$D_{stratified}$	0.018	0.001	0.009
$(C*h)^{\sim}_{\nearrow}$	$D_{equitoken}$	0.014	0.003	0.008
$C^{\{50\}}_{\nearrow}$	$D_{equitoken}$	0.012	0.002	0.007
C_{PPL}	$D_{equitoken}$	0.011	0.003	0.007
$C_{\nearrow}$	$D_{equitoken}$	0.011	0.002	0.007
$C^{\sim}_{\searrow}$	$D_{equitoken}$	0.012	0.002	0.007
$C^{E}_{\searrow}$	$D_{equitoken}$	0.012	0.002	0.007
C_{MATTR}	$D_{equitoken}$	0.011	0.002	0.007
E_{gpt2}	ext	0.001	0.000	0.001

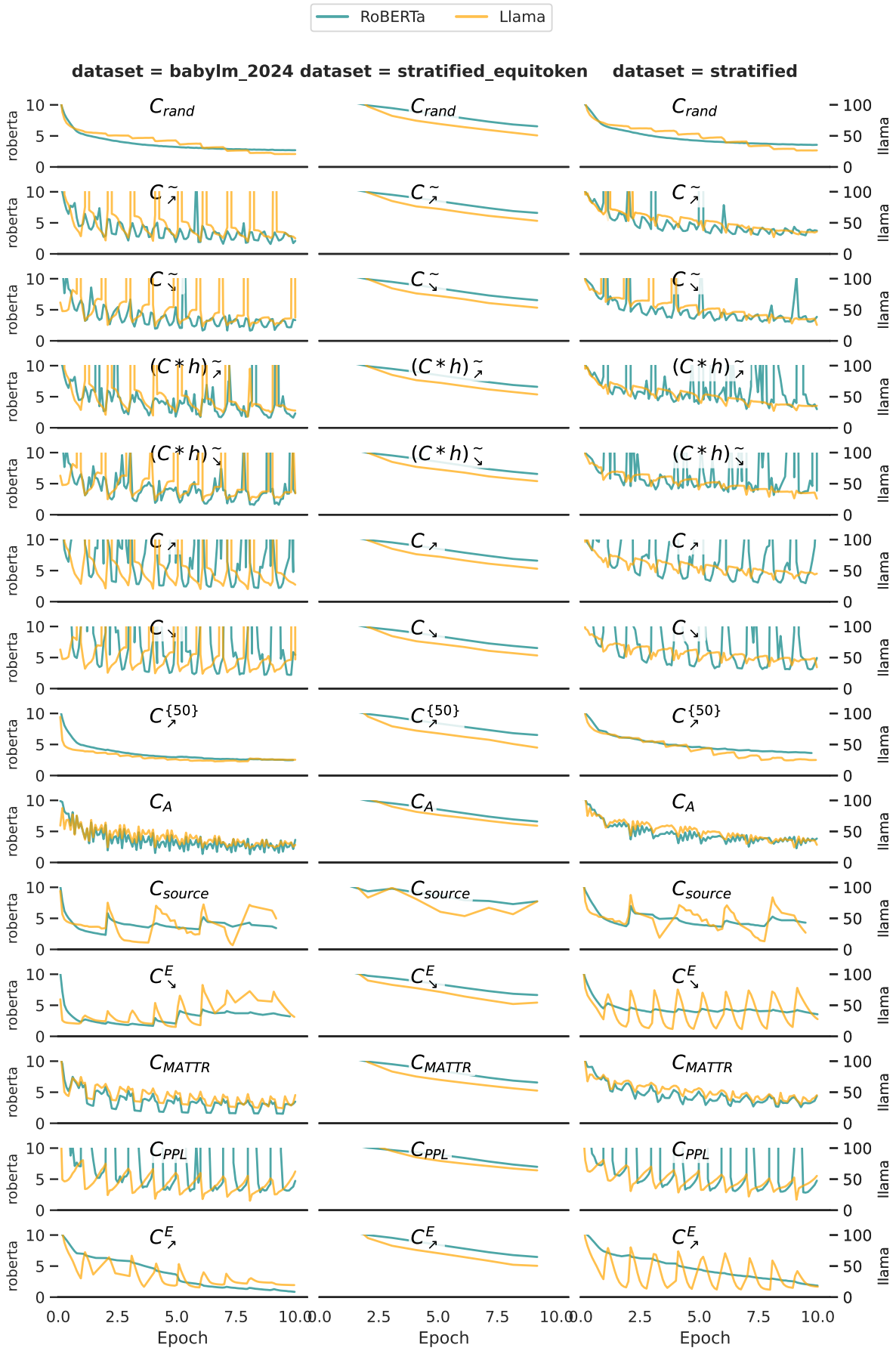


Figure 12: Training loss trajectories under different curricula.

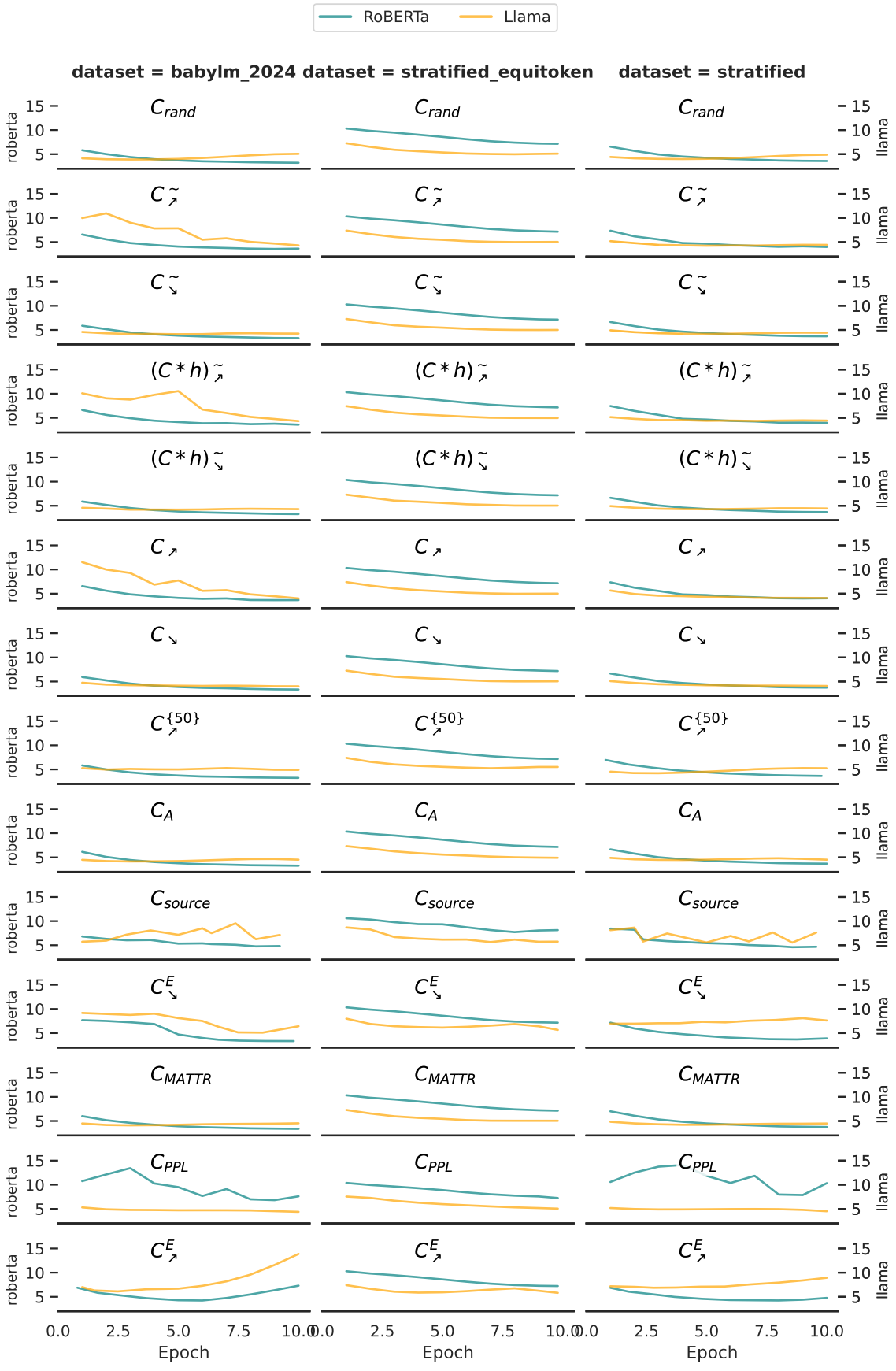


Figure 13: Evaluation loss trajectories under different curricula. We construct an evaluation set by sampling the 100M word 2024 BabyLM dataset (D_{2024} is the 10M version; Choshen et al., 2024). $|D_{eval}| = 0.05 \cdot |D_{2024}|$.

Understanding and Enhancing Mamba-Transformer Hybrids for Memory Recall and Language Modeling

Hyunji Lee[U*] **Wenhao Yu**[T] **Hongming Zhang**[T] **Kaixin Ma**[T]
Jiyeon Kim[K] **Dong Yu**[T] **Minjoon Seo**[K]

[U] UNC Chapel Hill [T] Tencent AI Lab [K] KAIST AI

Abstract

Hybrid models that combine state space models (SSMs) with attention mechanisms have shown strong performance by leveraging the efficiency of SSMs and the high recall ability of attention. However, the architectural design choices behind these hybrid models remain insufficiently understood. In this work, we analyze hybrid architectures through the lens of memory utilization and overall performance, and propose a complementary method to further enhance their effectiveness. We first examine the distinction between sequential and parallel integration of SSM and attention layers. Our analysis reveals several interesting findings, including that sequential hybrids perform better on shorter contexts, whereas parallel hybrids are more effective for longer contexts. We also introduce a data-centric approach of continually training on datasets augmented with paraphrases, which further enhances recall while preserving other capabilities. It generalizes well across different base models and outperforms architectural modifications aimed at enhancing recall. Our findings provide a deeper understanding of hybrid SSM-attention models and offer practical guidance for designing architectures tailored to various use cases. Our findings provide a deeper understanding of hybrid SSM-attention models and offer practical guidance for designing architectures tailored to various use cases[1].

1 Introduction

Recent advances in state-space models (SSMs), such as Mamba (Gu and Dao, 2023), have shown strong performance in language modeling, particularly in long-context tasks, while offering significantly greater efficiency than traditional Transformer (Vaswani et al., 2017) architectures (Dao and Gu, 2024; Waleffe et al., 2024; Zuo et al., 2024). However, unlike Transformers, which maintain a dynamically growing key-value (KV) cache to attend to all previous tokens, SSMs compress past information into a fixed-size hidden state, limiting their ability to model long-term dependencies and recall distant context (Park et al., 2024; Glorioso et al., 2024). To address this, recent work has explored *hybrid architectures* (Dong et al., 2024; Ren et al., 2024; Park et al., 2024) that integrate attention with SSMs, aiming to leverage the strengths of both: combining the expressive, high capacity memory of attention with the efficiency of SSM computation.

Despite promising results, there remains a limited understanding of how different architectural design choices affect performance in these hybrid models, and what specific roles SSM and attention components play. In this work, we aim to fill this gap by systematically analyzing the following three research questions: (*RQ1*) **Aggregation Strategies**: How do different ways of combining SSMs and attention affect performance and efficiency? (*RQ2*) **Component Roles**: What are the respective contributions and characteristics of SSMs and attention layers in hybrid models? (*RQ3*) **Data-Centric Enhancements**: Can performance be further improved through data-centric methods, beyond architectural design alone?

To investigate the first two questions (*RQ1*, *RQ2*), we conduct extensive pretraining experiments on 17 models spanning pure SSMs, Transformer, and hybrid variants (Figure 1). Prior work often uses inconsistent training and evaluation setups, making fair comparison difficult. We therefore design a unified experimental setup that standardizes training and evaluation, enabling a controlled analysis of individual components and architectural choices. All models share the same configurations, differing only in their core block design (SSM or attention). We evaluate them across three axes: language modeling, commonsense rea-

[*] Work was done during internship at Tencent AI Lab, Bellevue.

[1] Code in mamba-transformer-hybrids

Proceedings of the First BabyLM Workshop, pages 392–410
November 8, 2025

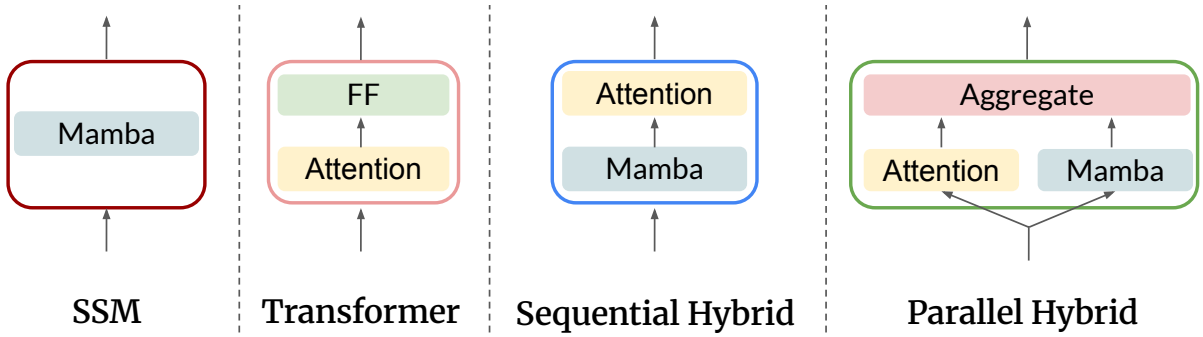


Figure 1: Comparison of different architectural designs: SSM, Transformer, Sequential Hybrid, and Parallel Hybrid. Each architecture consists of stacked *blocks* that incorporate Mamba and Attention layers. The key difference lies in how these layers are arranged: SSM uses only Mamba layers, Transformer uses only Attention layers, while the hybrid models combine both. Sequential Hybrid stacks Mamba and Attention layers within each block, whereas Parallel Hybrid applies them in parallel and aggregates their outputs. Feedforward (FF) layers are omitted in the hybrid models for clarity, as it varies by design.

soning, and memory recall. Our analysis shows a strong correlation between long-context language modeling and commonsense reasoning, but weaker links to memory recall. These results suggest that focusing solely on language modeling or reasoning benchmarks, as in prior work (Glorioso et al., 2024; Lieber et al., 2024; Ren et al., 2024), may miss critical aspects of memory performance. *Our study fills this gap by providing a comprehensive and standardized evaluation.*

Using our unified evaluation, we analyze *how aggregation strategies (sequential or parallel) affect performance and the roles of SSM and attention components*. Sequential hybrids, where one component processes input before the other, excel on short-context tasks because aligned representation spaces promote stable training. However, this alignment can limit expressiveness. In contrast, parallel hybrids keep separate embedding spaces and fuse outputs later, enabling greater representational diversity and stronger long-context performance. Among them, the parallel variant with a merge-attention layer, which attends over the outputs of the Mamba and the attention layers to produce a fused representation, achieves the strongest overall results.

Beyond architecture, we explore a *data-centric approach* to improve memory recall (*RQ3*). While previous works often finetune models on synthetic tasks like Needle-in-a-Haystack (NIAH) (Kamradt, 2023), which boosts recall but often harms performance on other metrics. To mitigate this, we show that continued training with paraphrased sentences, drawn from a distribution similar to the pretraining data, enhances recall with minimum or no degradation in commonsense reasoning. Compared to other datasets such as UltraChat (Ding et al., 2023), Based (Arora et al., 2024), or NIAH, this strategy achieves the best trade-off. Notably, it outperforms architectural methods aimed at enhancing recall, such as DeciMamba (Ben-Kish et al., 2024) (+12.7 avg), and generalizes well across a range of base models, scaling up to 2.8B parameter model.

2 Preliminary

In this section, we share details of how the Mamba layer from recent SSM models and the Attention layer in the Transformer differ, an overview of prior works on hybrid models, and outline our experimental architectures.

Mamba and Attention layers Both Mamba and attention layers transform an input sequence into an output sequence using a transformation matrix, but differ in how they process inputs. Mamba layers update a recurrent hidden state sequentially, incorporating one token at a time as a compressed summary of past inputs. In contrast, attention layers process the entire sequence simultaneously, attending to all preceding tokens to model dependencies. These approaches involve trade-offs: Mamba offers linear-time computation but may struggle with long-range dependencies, while attention layers capture such dependencies more effectively at the cost of quadratic time and memory. See Appendix A.1 for details and equations.

Hybrid Models To leverage the strengths of SSMs and attention, recent works have proposed hybrid architecture that integrate both components (Dong et al., 2024; Ren et al., 2024; Park

et al., 2024). These models outperforms non-hybrid models, especially in long-context language modeling compared to attention-only models and recall performance compared to pure SSMs.

Recent hybrid models vary along four design axes: (1) SSM layer type: Mamba is the most common (Gu and Dao, 2023; Ren et al., 2024; Dong et al., 2024; Glorioso et al., 2024), though alternatives like DeltaNet have also been effective (Yang et al., 2025). (2) Layer ratio: A 1:1 SSM-to-attention ratio is typical (Dong et al., 2024; Ren et al., 2024), though some prefer more SSM layers for efficiency (Glorioso et al., 2024; Lieber et al., 2024). (3) Attention type: To retain efficiency, many use SWA[2](Ren et al., 2024; Yang et al., 2025), combine SWA with full attention(Dong et al., 2024), or use full attention alone (Glorioso et al., 2024). (4) Integration strategy: Sequential fusion is most common (Park et al., 2024; Ren et al., 2024; Yang et al., 2025), but parallel fusion is also explored (Dong et al., 2024).

In this work, as our focus is on understanding affect of how to combine SSMs with attention layers and analyzing the role of each components in hyrid model performance, we focus on the fourth axes and keep other design choices fixed based on the recent strong baselines (Ren et al., 2024; Dong et al., 2024; Yang et al., 2025): (1) using Mamba as the SSM component, (2) a 1:1 ratio of attention to SSM layers, and (3) using SWA (Beltagy et al., 2020) as attention layer. See Appendix B for more related works.

Architectural Designs of Hybrid Blocks To analyze various hybrid model configurations, we design a set of *hybrid models*, each combining Mamba and Attention layers. These blocks are stacked to build the full model (Figure 1). Our designs vary along two main axes: (1) the integration strategy and (2) the placement of feed-forward (FF) layers. For integration, we explore: **sequential hybrid** where one layer's output feeds into the other, with two variants (Mamba → SWA and SWA → Mamba) and **parallel fusion** where both layers receive the same input, and their outputs are aggregated using one of several methods (simple averaging (Dong et al., 2024), a trainable projection layer (Behrouz et al., 2024), or a trainable merge-attention layer). Also, given that FF layers play an important role in Transformer models (Geva et al., 2020; Meng et al., 2022), we also experiment with the effect of different FF placements.

[2]SWA restricts attention to a fixed-size window around each token, improving scalability over full attention.

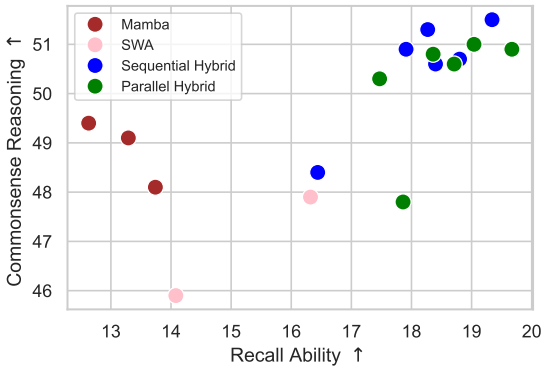


Figure 2: Comparison of different model architectures on Commonsense Reasoning (y-axis) vs. Recall Ability (x-axis). Commonsense Reasoning and Recall Ability are measured using answer accuracy. The models compared included Mamba-only, SWA-only, Hybrid (Sequential), and Hybrid (Parallel). For details of each model, see Figure 10 in Appendix C.1.

3 Designing a Unified Experimental Setup

While various works have proposed and demonstrated the effectiveness of hybrid models, their results are often difficult to compare to each other due to differences in training procedures, evaluation metrics, and the absence of released checkpoints. To enable fair and comprehensive analysis, in this section, we introduce a unified experimental setup to re-evaluate multiple models within this consistent framework of training (Section 3.1) and evaluation (Section 3.2). We observe that some prior works often overlook key metrics, which can obscure a model's overall performance, underscoring the need for extensive evaluation over multiple axes to understand model performance.

3.1 Training

We follow widely adopted training setups from recent works, primarily based on Ren et al. (2024), which provides detailed implementation code. All models are trained from scratch on 100B tokens from the SlimPajama dataset (Soboleva et al., 2023). Model sizes are kept consistent across architectural variants: approximately 430M parameters for base models and 1.3B for larger ones. All models use the same hyperparameters: batch size of 512, sequence length of 4K, learning rate of 4e-4, weight decay of 0.1, window size of 2k for SWA, and the AdamW optimizer (Loshchilov and Hutter, 2017).

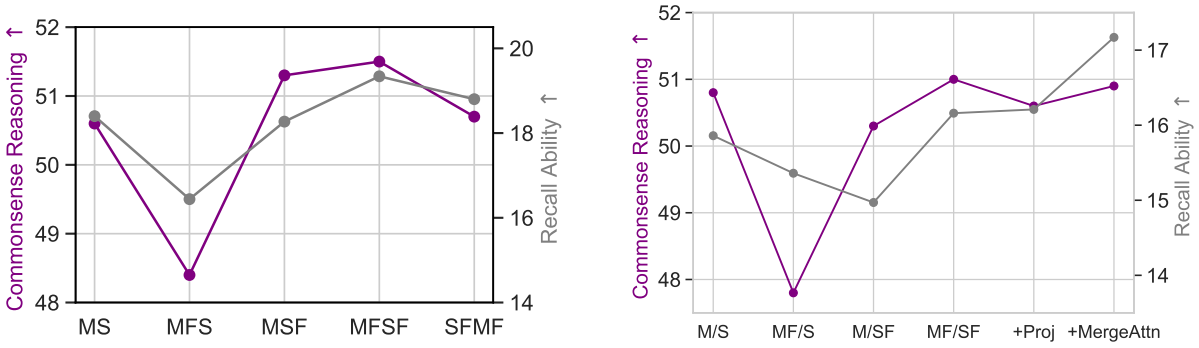


Figure 3: Performance comparison of commonsense reasoning accuracy and recall ability across different model architectures. (**a**) Results for sequential models. (**b**) Results for parallel models. For further details, refer to the first paragraph of Section 4.1.

3.2 Evaluation

Setup We evaluate hybrid models across three axes: (1) long-context language modeling, (2) commonsense reasoning, and (3) memory recall, following previous works on hybrid models. For language modeling, we report perplexity on the SlimPajama validation set using 16k-token sequences. Commonsense reasoning is assessed by averaging accuracy across five standard benchmarks: LAMBADA-OpenAI (Radford et al., 2019), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), ARC-Easy (Clark et al., 2018), and Winogrande (ai2, 2019). Recall ability is evaluated over average of eight datasets in Based benchmark (Arora et al., 2024), using the evaluation protocol of Yang et al. (2025). We further group them into short- and long-context subsets to study the influence of context length on recall performance. Details are in Appendix C.2.

Correlation Between Evaluation Axes We investigate how the three evaluation axes, language modeling, commonsense reasoning, and memory recall, relate across different architectural choices. We find that **strong performance on reasoning or language modeling does not necessarily imply strong memory recall.** While there is some positive correlation, it is relatively weak. Specifically, the pearson correlation coefficient between language modeling and commonsense reasoning is high (0.814), whereas recall correlates modestly with reasoning (0.697) and even less with language modeling (0.542). These trends are also visualized in Figure 2, which shows the correlation between recall ability (x-axis) and reasoning (y-axis). Notably, the clustering of models with similar architectures (indicated by color) suggests that architectural design has a greater impact on recall performance than overall reasoning ability. These findings highlight that prior works, which evaluate models solely on language modeling or reasoning benchmarks (Glorioso et al., 2024; Lieber et al., 2024; Ren et al., 2024), need a more comprehensive evaluation including memory-intensive tasks to more accurately assess model capabilities.

4 How Does the Architectural Design Affect Model Performance?

In this section, we present our experimental results across various model architectures (Section 4.1) and provide a detailed analysis of their structural design (Section 4.2).

4.1 Results

Figure 3 compares commonsense reasoning and recall performance across various *block* designs in both sequential and parallel model architectures. In both subfigures, the x-axis represents different block configurations. M indicates a Mamba layer, S a Sliding Window Attention (SWA) layer, and F a feed-forward (FF) layer. For example, MFSF represents a block with Mamba, FF, SWA, and FF layers in that order. In parallel models (Figure 3b), '|' denotes parallel branches (e.g., M|SF means Mamba on one side and SWA+FF on the other). Aggregation strategies are defined as follows: +PROJ uses a trainable projection layer; +MERGEATTN uses a trainable attention module, similar to the cross-attention layer in encoder-decoder models, but using Mamba's output embeddings as the Key and Value; the remaining variants use simple mean averaging. See Appendix D.1 for detailed performance and Appendix D.2 for com-

parison between hybrid and non-hybrid models.

Impact of SWA and Mamba Layer Order on Sequential Hybrid Performance We investigate how the order of SWA and Mamba layers affect sequential hybrid performance by comparing two configurations: MFSF (Mamba before SWA) and SFMF (SWA before Mamba). As shown in Figure 3a, MFSF consistently outperforms SFMF across tasks. This suggests that placing the Mamba layer first helps the model capture global dependencies early, while placing SWA first may bottleneck performance due to its limited attention window. However, when analyzing recall performance by context length (Figure 16 in Appendix D.3), SFMF performs better on shorter contexts. We attribute this to SWA effectively approximating full attention when the input length is within its window, enabling strong local representations that Mamba can refine. In summary, SFMF may benefit short-context tasks, but MFSF, architecture used in Ren et al. (2024), offers superior overall performance. We therefore adopt MFSF as our default sequential model architecture.

Effect of Aggregation Method in Parallel Hybrids Performance We study how different aggregation layers for combining SWA and Mamba output embeddings affect hybrid model performance. As shown in Figure 3b, we evaluate three strategies: +BOTH, PROJ, and MERGEATTN. MERGEATTN achieves the best overall performance, particularly in long-context language modeling (see Appendix D.4 for more results). We thus use MERGEATTN as the representative parallel model in subsequent analysis. Simple averaging (+BOTH) performs well on commonsense reasoning, consistent with observation in Dong et al. (2024), but we observe that it underperforms on recall; for strong recall, especially with long contexts, trainable aggregation methods like PROJ and MERGEATTN are more effective.

Sequential models excel in short contexts, parallel models excel in long ones When comparing recall performance of sequential and parallel models, we observe that sequential models tend to perform better in relatively shorter contexts whereas parallel combinations general show superior performance in longer contexts (Figure 4). We hypothesize that this trend arises from the differing degrees of interaction between the SWA and Mamba components. As parallel models has less interaction

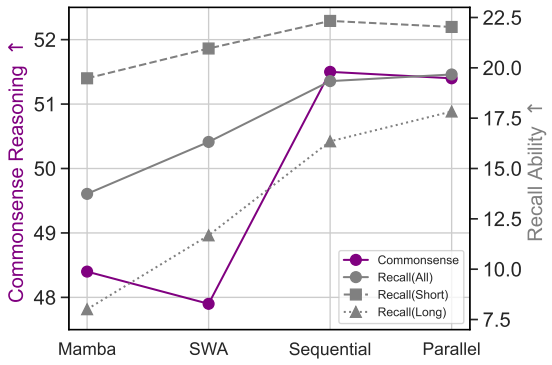


Figure 4: Performance of best performing models from each architecture in commonsense reasoning and recall ability, where divided by length of context.

between Mamba and SWA components, it prevents from collapsing into a shared mode of producing overly similar hidden states. It instead encourages each component to retain its distinct representational strength. In Section 4.2, we provide empirical evidence supporting this hypothesis.

Impact of Adding Feed-Forward Layers on Hybrid Model Performance Feed-formward (FF) layers play an important role in transformers (Geva et al., 2020; Meng et al., 2022), but their effect on hybrid models remains less explored. We find that adding FF layers to only one component, either Mamba or SWA, degrades performance in both sequential and parallel settings, while improvements appear only when FF layers are added to *both* components. We hypothesize that this degradation arises from feature misalignment: it is especially harmful in parallel architectures, where components maintain distinct representations and make aggregation harder, whereas sequential models integrate features into a shared space, mitigating some of these issues. This drop is particularly high when adding FFNs to Mamba, likely because its final layer (C in Equation 1) already functions similarly to an MLP (Sharma et al., 2024), making additional FFNs redundant or even detrimental. This aligns with prior findings that FFNs benefit SWA but not Mamba (Gu and Dao, 2023).

Generalization to 1.3B Trends observed at the 430M scale generally hold at 1.3B. Hybrid models consistently outperform non-hybrids. Among sequential hybrids, MFSF outperforms MS. In parallel setups, merge-attention as an aggregation layer shows higher performance, especially for long-context recall. Overall, merge-attention mechanisms show strong performance. Sequential hy-

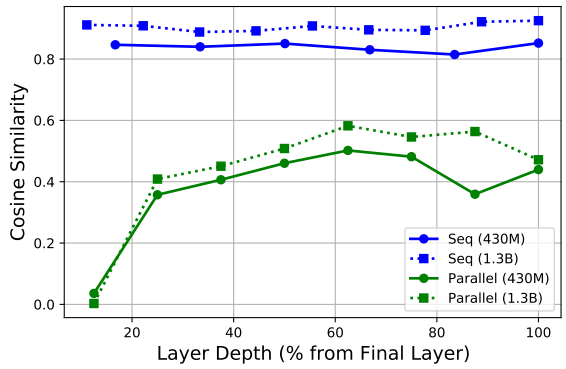


Figure 5: Cosine similarity between output embeddings of aligned SWA and Mamba layers (y-axis), plotted against layer depth, measured as percentage distance from the final layer (x-axis).

brids excel in short-context settings, while parallel hybrids perform better with longer contexts. See Appendix D.6 for detailed results.

4.2 Analysis

Similarity between SWA and Mamba Output Embeddings in Hybrid Models To better understand the interaction between SWA and Mamba in hybrid models, we analyze the cosine similarity of their output embeddings across block depths, aligned by position from the final block (Figure 5). Sequential hybrids show high similarity, especially in the larger 1.3B model, because outputs from one component feed into the next, naturally aligning their representations. Parallel hybrids show much lower similarity, particularly in early and middle layers, as both components process inputs independently and fuse outputs later. We hypothesize that this structural difference shapes performance: sequential hybrids benefit from stable, aligned representations for commonsense reasoning and short-context tasks but struggle with long-context reasoning. In contrast, parallel hybrids produce more diverse representations and, though sensitive to aggregation strategy, can outperform on complex long-context tasks when effectively trained. More analysis in Appendix D.7.

Identifying Critical Components in Hybrid Blocks Figure 6 shows performance degradation on commonsense (left) and recall (right) tasks when removing blocks by depth. Removing the first block causes the steepest drop, up to 90% on recall tasks, highlighting the crucial role of early layers. We further examine the importance of subcomponents within each block. In sequential models, the first subcomponent is most critical because it shapes the feature space, and later components align to it. In parallel models, the aggregation layer is most critical as it must merge the distinct representation spaces from Mamba and SWA, while either path alone can still infer the input distribution. See Appendix D.8 for further discussion.

Understanding the performance gains of MERGEATTN Among the various configurations, parallel hybrids using an attention layer that merges output embeddings of SSM and attention achieve the best performance. To understand why these models tend to perform strongly, we analyze the models on how much each token is influenced by prior tokens, following the method in Ben-Kish et al. (2024); higher value indicates that they exhibit stronger attention to previous tokens. As shown in Figure 7, models with merge-attention show the highest average attention weights, suggesting that their improved performance arises because the Mamba layers effectively capture global dependencies, which the merge-attention mechanism then leverages to integrate information. See Appendix D.9 for more detail of calculation.

5 Dataset Strategy to Enhance Recall

We show that continually training models on datasets with paraphrased contexts, drawn from a distribution similar to the pretraining dataset, improves recall without sacrificing commonsense reasoning. Previous work focused on improving recall through *architectural changes*, such as hybrid models. Here, we investigate a *data-centric approach*, aiming to complement and further enhance these architectural advances.

Section 5.1 describes how we construct the training dataset and train the model. Section 5.2 shows that models trained on our dataset achieve the best trade-off between recall and reasoning, outperforming other dataset choices and DeciMamba (which introduces architectural changes) across scales up to 2.8B parameters. Section 5.3 analyzes design factors such as input length, dataset size, and model choice, demonstrating through extensive experiments that our simple approach generalizes well and consistently improves performance.

5.1 Experimental Setup

Paraphrasing Method We construct a paraphrased dataset using a subset of the training corpus (SlimPajama), based on the hypothesis that

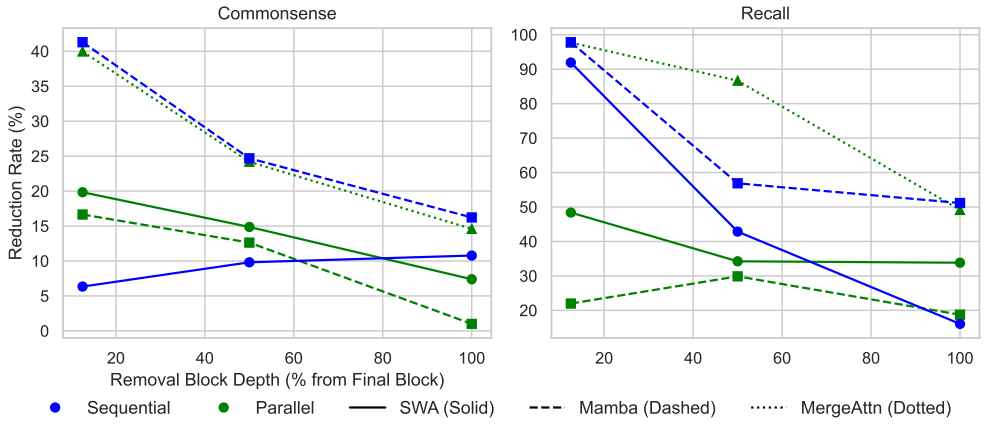


Figure 6: Performance degradation (y-axis) on commonsense (left) and recall (right) tasks as a function of the removed block's relative position from the final block (x-axis).

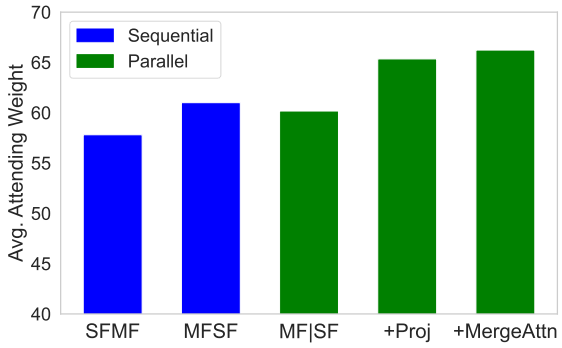


Figure 7: Average attending weight across different model architectures. Higher values indicate that the model attends more strongly to previous information.

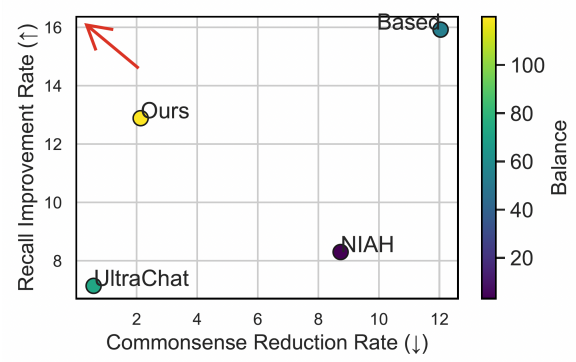


Figure 8: The upper-left region (indicated by the red arrow) represents the optimal balance between recall improvement and commonsense degradation.

the data should remain close in distribution to the original pretraining corpus to prevent degrading existing performance. To control the density of paraphrased content, we divide the data into 1k-token chunks. For each chunk, we use LLaMA 3.1–8B[3] to generate factual question-answer (QA) pairs. Following Arora et al. (2024), we convert these QA pairs into cloze-style paraphrased sentences. This yields pairs of the form (1k-token chunk, paraphrased sentence). To construct a training dataset, we concatenate multiple chunks and insert the corresponding paraphrased sentence at a random position following the chunk it was derived from. Based on the constructed dataset, we run a filtering process based on three criteria: (1) the model fails to generate a valid question and answer pair, (2) the generated answer is not present in the corresponding paragraph, or (3) the model fails to convert the example into a cloze-style task. See Appendix E.1 for more details.

[3]We use the released model from Hugging Face: `meta-llama/Meta-Llama-3-8B-Instruct`

Training Details After the initial pretraining phase,[4] we continue training the model using several different datasets, including recall-intensive datasets such as NIAH and SQuAD from Based, widely used SFT dataset UltraChat (Ding et al., 2023), and our paraphrased dataset. Following the setup of Ben-Kish et al. (2024), we train the models using a batch size of 32, a learning rate of 1e-4 for 10 epochs. We conduct experiments with both hybrid models and Mamba-only models.

5.2 Results

Our Dataset Strikes the Best Balance Figure 8 shows the balance between commonsense degradation and recall gains[5] for the 430M sequential hybrid model (MFSF) when trained on various datasets including NIAH (Kamradt, 2023), UltraChat (Waleffe et al., 2024), or SQuAD dataset from

[4]We also experimented with incorporating the paraphrase dataset during pretraining. However, we observed a degradation in performance when doing so (see Appendix E.2).

[5]In this section, we exclude SQuAD from Based when computing average recall, as it is part of the training data.

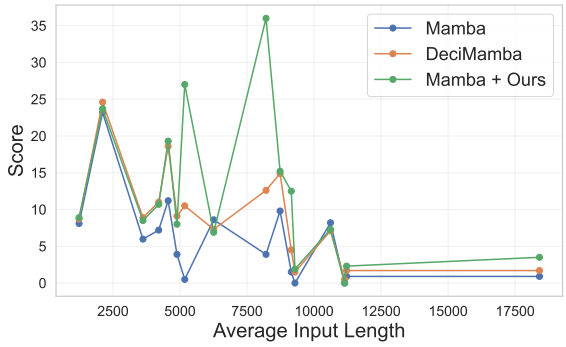


Figure 9: Performance (y-axis) of Mamba, DeciMamba, and Mamba trained with our dataset across LongBench datasets with varying input lengths (x-axis).

Based benchmark (Arora et al., 2024). Models trained on our paraphrased SlimPajama dataset consistently achieve the best balance. We attribute this to: (1) its alignment with the original pretraining distribution, preserving baseline performance; and (2) paraphrased content promoting the model to retain and utilize previous context. In contrast, recall-focused datasets like NIAH and Based significantly harm commonsense performance, while UltraChat offers only modest recall improvements. Appendix E.3 provides detailed performance. Similar patterns hold for Mamba models, including the released 2.8B version (Appendix E.4).

Comparison with DeciMamba We investigate whether a *data-centric* approach can outperform architectural modifications by comparing our approach with DeciMamba (Ben-Kish et al., 2024), which enhances recall by discarding less important tokens. Across 16 datasets in LongBench (Bai et al., 2023) using Mamba-2.8B, our approach achieves an average of +12.7 points overall, with particularly strong gains in QA tasks (+8.1 on average). As shown in Figure 9, our approach tends to consistently outperform DeciMamba on medimum and long input lengths. These results suggest that our data-centric approach is not only complementary to architectural change but can also show strong standalone performance. Full results are provided in Appendix E.5.

5.3 Analysis

To understand the benefit and affect of such approach, we analyzed over various design choices.

Generalizes to Various Base Models We observe that **our method generalizes across different released variants of Mamba-2.8B**. Continual training yields performance gains for the base model (+1.7 in commonsense, +6.5 in recall), as well as for instruction-tuned (+1.3 in commonsense, +3.6 in recall) and preference-aligned models (+3.4 in commonsense, +0.8 in recall). Improvements are generally more pronounced for the base model. See Appendix E.6 for model details and results.

Longer chunk sizes yield stronger results We observe that models trained on longer sequences tend to achieve lower reduction rates on commonsense tasks and substantially higher gains on recall tasks, especially on long-context tasks (Figure 19 in Appendix)[6]. Training on shorter chunk sizes (e.g., 2k) tends to enhance performance on short-context recall but leads to high degradation on long-context tasks. This trend is robust across architectures, appearing in both sequential and parallel hybrids, and holds for different model sizes. For detailed results, refer to Appendix E.7.

Performance improves as the size of the training dataset increases We observe that training with larger datasets leads to clear gains in both commonsense reasoning and recall tasks: models trained with more tokens achieve a lower reduction rate on commonsense benchmarks and steadily higher improvements on recall performance[7]. Performance grows with the amount of training data and begins to converge around 80M-100M tokens. This trend holds across different model sizes and hybrid architectures. More details are in Appendix E.8.

6 Conclusion

In this paper, we focus on two main aspects: (1) studying how different architectural design choices (sequential, parallel) affect performance in hybrid models and the roles of individual components (SSM and Attention layers), and (2) exploring data-centric approaches to further improve model's recall ability. Our findings show that sequential models offer stable training but are limited in expressiveness, while parallel architectures better preserve the unique characteristics of each component, often leading to stronger performance. In particular, parallel hybrid models with merge-attention-based aggregation consistently perform well. We also demonstrate that continually pretraining the model on a paraphrased dataset effectively improves recall while maintaining overall model performance.

[6] All experiments used a fixed token count of 10M, discarding the final chunk if it does not align with the sequence length.

[7] All experiments use a fixed chunk size of 4k

Limitations

Due to computational constraints, we conducted our experiments on relatively small model scales, 430M and 1.3B parameters, trained with 100B tokens. Pretraining a 430M-parameter model on 8 A100 GPUs takes about one week, while a 1.3B-parameter model requires roughly two weeks, making it challenging to analyze larger-scale models. Notably, prior work on hybrid models has also primarily operated at similar scales (Ren et al., 2024; Dong et al., 2024; Yang et al., 2025). These resource limitations also restricted our ability to explore a broader range of hybrid model configurations and focus on the experimental setup described in the "Hybrid Model" section (Section 2). We leave more extensive analyses, such as incorporating additional components like gated DeltaNet, to future work.

References

2019. Winogrande: An adversarial winograd schema challenge at scale.

Simran Arora, Aman Timalsina, Aaryan Singhal, Sabri Eyuboglu, Xinyi Zhao, Ashish Rao, Atri Rudra, and Christopher Ré. 2024. Just read twice: closing the recall gap for recurrent language models.

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language models enable simple systems for generating structured views of heterogeneous data lakes. *Preprint*, arXiv:2304.09433.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. 2024. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. 2024. Decimamba: Exploring the length extrapolation potential of mamba. *Preprint*, arXiv:2406.14528.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. 2024. Hymba: A hybrid-head architecture for small language models.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Paolo Glorioso, Quentin Anthony, Yury Tokpanov, James Whittington, Jonathan Pilault, Adam Ibrahim, and Beren Millidge. 2024. Zamba: A compact 7b ssm hybrid model. *Preprint*, arXiv:2405.16712.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *COLM*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Gregory Kamradt. 2023. Needle in a haystack- pressure testing llms. *Github*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Hyunji Lee, Sejune Joo, Chaeeun Kim, Joel Jang, Doyoung Kim, Kyoung-Woon On, and Minjoon Seo. 2023. How well do large language models truly ground? *arXiv preprint arXiv:2311.09069*.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi,

Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, and 1 others. 2024. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*.

Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. 2019. OpenCeres: When open information extraction meets the semi-structured web. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3047–3056, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Neural Information Processing Systems*.

Jongho Park, Jaeseung Park, Zheyang Xiong, Nayoung Lee, Jaewoong Cho, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. 2024. Can mamba learn how to learn? a comparative study on in-context learning tasks. *International Conference on Machine Learning*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Preprint*, arXiv:1806.03822.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. 2024. Samba: Simple hybrid state space models for efficient unlimited context language modeling. *ICLR*.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in mamba. *arXiv preprint arXiv:2404.03646*.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, and 1 others. 2024. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2025. Gated delta networks: Improving mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. 2024. Falcon mamba: The first competitive attention-free 7b language model. *arXiv preprint arXiv:2410.05355*.

A Preliminary

A.1 Mamba and Attention layers

Given an input sequence X, both Mamba layers and attention layers transform it into an output sequence Y via a transformation matrix M: M_{Mamba} (Equation 1) and M_{Attn} (Equation 2). The key difference lies in how they process inputs. Mamba layers update a recurrent hidden state h_t sequentially, incorporating one token x_t at a time. This hidden state serves as a compressed memory summarizing all past inputs. In contrast, Attention layers process the entire input sequence at once, attending to all tokens up to the current position, thereby capturing dependencies without recurrence. These design choices yield different trade-offs. Mamba is more computationally efficient due to its linear-time recurrence but may struggle with long-range dependencies. Attention layers, while effective at modeling token-wise relationships, incur quadratic time and memory complexity with sequence length.

$$Y_{\text{Mamba}} = M_{\text{Mamba}} X \quad \text{where} \tag{1}$$

$$Y_{\text{Mamba},t} = Ch_t, \quad h_t = Ah_{t-1} + Bx_t, \quad x_t \in X$$

$$Y_{\text{Attn}} = M_{\text{Attn}} X \quad \text{where} \tag{2}$$

$$Y_{\text{Attn}} = \text{softmax}\left(\frac{(W_Q X)(W_K X)^T}{\sqrt{d_k}}\right)(W_V X)$$

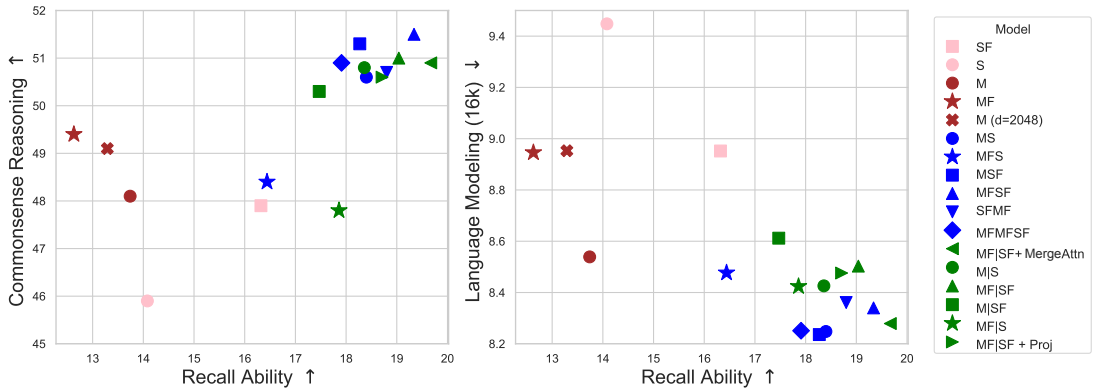


Figure 10: Comparison of detailed model architectures on Commonsense Reasoning (y-axis on left) and Language Modeling(y-axis on right) vs. Recall Ability (x-axis). Commonsense Reasoning and Recall Ability are measured using answer accuracy. The models compared included Mamba-only, SWA-only, Hybrid (Sequential), and Hybrid (Parallel).

B Related Works

B.1 Studies on SSMs

Prior work has explored state space models (SSMs) (Waleffe et al., 2024; Sharma et al., 2024), primarily focusing on their performance in language modeling tasks, particularly their ability to handle long-context dependencies. However, these studies typically examine pure SSM architectures and do not consider hybrid models. In this work, we conduct an empirical investigation of *hybrid architectures* that combine SSM and attention layers. Our goal is to understand the source of their performance gains and the distinct roles played by each component.

B.2 Recall Ability of Language Models

Enhancing a model's recall ability, also referred to as grounding ability, is a critical aspect of language modeling, especially in scenarios where the model must answer questions based on a given context, maintain strong coherence across parts of a conversation or document, or perform consistent reasoning over extended texts (Arora et al., 2024; Lee et al., 2023). This ability allows the model to retrieve relevant information accurately from given context, sustain contextual coherence, and generate factually grounded responses.

In this paper, we define recall ability as distinct from the general capability to model long contexts. Unlike next-token prediction, recall-intensive tasks require the model to retrieve specific values or answers from earlier in the context, demanding precise and accurate memory. Furthermore, evaluating recall ability is not limited to long-context tasks; it applies to any setting where exact retrieval from prior context is necessary.

Several studies have shown that SSM-based models often struggle with such recall-intensive tasks, as they must encode prior context into fixed-size hidden states. This architectural constraint leads to a bottleneck that limits their recall performance (Park et al., 2024; Ren et al., 2024; Dong et al., 2024).

C Designing a Unified Experimental Setup

C.1 Correlation Between Evaluation Axes

Figure 10 shows the detailed configuration of each points in Figure 2.

C.2 Dataset

We experiment over dataset from Arora et al. (2024) (Based benchmark) to calculate recall ability. Based benchmark is comprised of eight datasets: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), DROP (Dua et al., 2019), FDA (Arora et al., 2023), SWDE (Lockard et al., 2019), and SQuAD (Rajpurkar et al., 2018). The NQ dataset is further subdivided by input length into NQ-512, NQ-1024, and NQ-2048. To compare the recall ability across different sequence lengths, we categorize the eight datasets into two groups: relatively short sequences (NQ-512, DROP, TriviaQA, SQuAD) and relatively long sequences (NQ-1024, NQ-2048, FDA, SWDE). Using the LLaMA-2 tokenizer (Touvron et al., 2023), which was also used during training, the average input length is around 1k tokens for the short-sequence group and around 2.5k tokens for the long-sequence group.

The Slimpajama dataset and evaluation datasets are released under Apache 2.0 license. We used the datasets for research purpose.

D How does the architectural design affect model performance?

D.1 Performance at the 430M scale

Table 1 presents the performance of models at a 430M parameter scale. Figures 11 and 12 show the language modeling performance, measured in terms of perplexity on the SlimPajama validation set, for sequential and parallel hybrid models, respectively.

D.2 Hybrid models outperform non-hybrid models in recall and commonsense reasoning

Results in Figure 4 show the performance of non-hybid models (Mamba, SWA) and hybrid models (Sequential, Parallel). We observe consistent gains in both commonsense reasoning and recall performance in hybrid models over non-hybrid ones in line with prior works (Ren et al., 2024; Dong et al., 2024; Park et al., 2024). Notably, we observe that recall shows a substantially larger improvement, with an average increase of 29.5%, compared to a 7.3% gain in commonsense reasoning. These improvements are especially prominent in long-context scenarios. In contrast, in short-context settings, performance differences are less pronounced, and hybrid models perform similarly to the SWA baseline.

D.3 SWA as the Initial Component Improves Short Context Recall

Figure 16 presents the average recall performance for both short and long sequences in the sequential architecture. The SFMF configuration demonstrates stronger performance on shorter sequences. We hypothesize that this is because, in short contexts, the input length fits within the window size of the SWA module, allowing it to approximate full attention more effectively.

D.4 Trainable Aggregation Layers Improve Performance on Long Contexts

Figure 12 presents perplexity on the SlimPajama validation dataset across different chunk sizes. Models equipped with trainable aggregation layers, specifically MF|SF (+proj) and

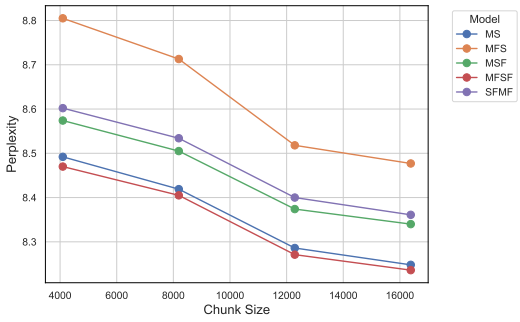


Figure 11: Peplexity over sequence length for sequential hybrids

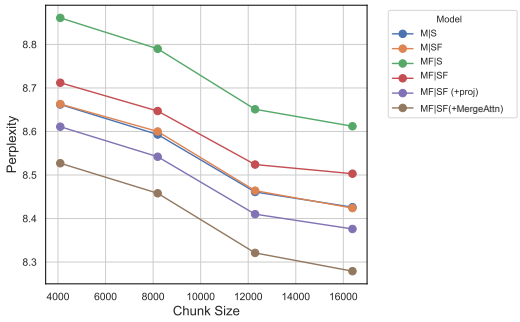


Figure 12: Peplexity over sequence length for parallel hybrids

MF|SF (+MergeAttn), consistently outperform others across varying context lengths. These models show strong recall performance, with particularly notable improvements in longer ones (Figure 13).

D.5 Impact of Adding Feed-Forward Layers on Hybrid Model Performance

Prior work has shown the importance of feed-forward (FF) layers in transformers (Geva et al., 2020; Meng et al., 2022). Thereby, we investigate their impact on hybrid models. Interestingly, adding FF layers to only one component, either Mamba or SWA, degrades performance in both sequential and parallel settings, and performance improves only when FF layers are added to both components. In the sequential setup (Figure 3a), the baseline MS outperforms MFS and MSF, but is lower than MFSF. Similarly, in the parallel setup (Figure 3b), M|S show higher performance over MF|S and SF|M but lower than SF|MF.

We hypothesize that this degradation stems from feature misalignment. The effect is more pronounced in parallel architectures, where individual component characteristics are preserved, making it harder to aggregate misaligned features. In contrast, sequential models integrate features into a shared

	Commonsense Reasoning						Recall Ability								
Model Type	LAM.	Hella.	PIQA	ARC	Wino.	***Avg.***	NQ-S	NQ-M	NQ-L	Drop	FDA	SWDE	TQA	SQD	***Avg.***
M	31.7	43.1	68.2	45.2	52.6	48.1	9.6	8.7	7.5	11.4	1.8	14.1	38.5	18.4	13.7
MF	34.2	41.6	68.6	51.8	51.0	49.4	9.2	8.8	6.3	10.4	1.1	12.3	36.0	17.0	12.6
S	30.6	37.6	64.6	45.3	51.3	45.9	9.9	9.2	6.4	12.4	3.2	18.4	31.9	13.4	13.1
SF	35.4	38.7	65.7	48.7	51.0	47.9	10.8	7.9	6.8	12.3	15.5	16.5	40.8	20.0	12.6
Sequential Hybrid															
MS	40.8	44.0	67.2	50.0	50.9	50.6	12.6	12.2	8.0	11.6	14.3	26.4	41.7	20.6	18.4
MFS	34.9	41.8	67.6	44.5	53.2	48.4	10.1	8.8	7.6	11.8	14.6	21.6	37.8	19.3	16.4
MSF	39.0	43.3	67.9	52.5	53.8	51.3	12.3	11.5	7.0	11.7	16.4	24.8	41.9	20.6	18.3
MFSF	38.5	44.2	69.1	51.7	54.0	**51.5**	13.5	12.3	8.0	11.2	16.5	28.6	43.4	21.2	19.3
SFMF	37.4	42.8	68.6	52.1	52.5	50.7	12.8	11.6	7.6	12.2	15.5	25.6	43.2	22.0	18.8
Parallel Hybrid															
M|S	40.1	42.7	67.9	50.1	53.1	50.8	11.5	10.5	7.3	11.9	16.2	26.7	42.8	20.0	18.4
MF|S	24.5	42.5	68.3	52.3	51.7	47.8	11.3	11.3	7.2	11.6	15.5	25.6	40.9	19.6	17.9
M|SF	37.5	41.2	67.6	51.4	53.7	50.3	11.0	10.0	6.9	10.9	14.9	24.5	41.7	19.9	17.5
MF|SF (Avg)	39.3	42.8	67.9	52.8	52.3	51.0	11.7	11.9	8.0	12.3	16.8	28.0	42.1	19.9	18.8
MF|SF (Proj)	38.0	42.6	69.4	51.0	52.0	50.6	11.9	12.4	8.4	12.7	16.9	28.6	42.3	20.7	19.2
MF|SF (MergeAttn)	39.3	44.3	69.0	51.9	52.3	51.4	12.8	12.9	9.0	11.9	17.7	29.6	43.1	20.3	**19.7**

Table 1: Model performance at the 430M scale. Model Type: M = Mamba, S = SWA, F = FF layer. The order reflects the design sequence within each block. In parallel hybrids, "|" denotes parallel branches (e.g., M|SF means Mamba on one side, SWA+FF on the other). Tasks: LAM. = LAMBADA-OpenAI, Hella. = HellaSwag, ARC = ARC-Easy, Wino. = Winogrande, NQ-S = NQ-512, NQ-M = NQ-1024, NQ-L = NQ-2048, TQA = TriviaQA, SQD = SQuAD. Bold indicates the highest average performance. In both cases, the best models use hybrid architectures with merge-attention.

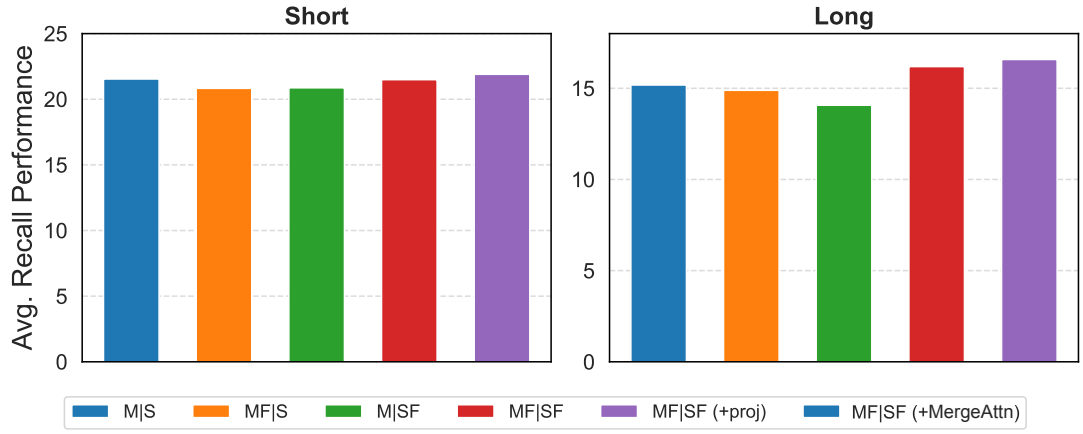


Figure 13: Comparison of average recall performance across short and long input contexts for parallel hybrids

space, mitigating this effect. Also, the performance drop is especially large when adding FFNs to the Mamba layer, possibly because its final layer (C in Equation 1) already functions similarly to an MLP (Sharma et al., 2024), making an additional FFN redundant or even detrimental. This aligns with prior findings that FFNs benefit SWA but not Mamba (Gu and Dao, 2023).

D.6 Performance at the 1.3B Scale

Table 2 presents the performance of models at the 1.3B parameter scale. Due to computational constraints, we limited our experiments to configurations that demonstrated strong performance at the 430M scale. We observe consistent trends across both scales. Hybrid models outperform their non-hybrid counterparts. Among sequential architectures, the MFSF model achieves the best performance. Additionally, parallel architectures that use merge-attention layers for fusion generally yield the highest performance. We also observe a similar pattern from 430M scale (Figure 4) when comparing short- vs. long-sequence settings in recall ability in 1.3B scale (Figure 15).

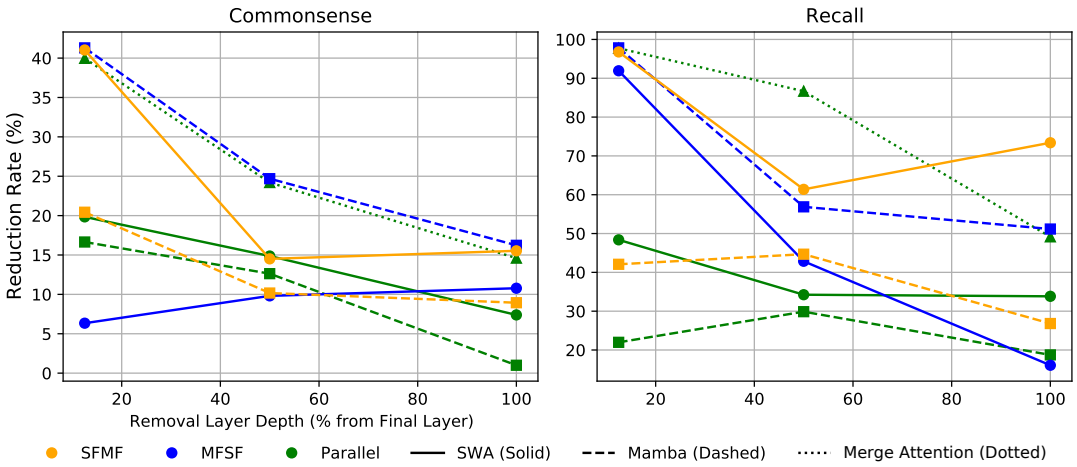


Figure 14: Performance degradation (y-axis) on commonsense (left) and recall (right) tasks as a function of the removed block's relative position from the final block (x-axis) for sequential(SFMF), sequential(MFSF) and parallel(+MergeAttn) architecture.

	Commonsense Reasoning						Recall Ability								
Model Type	LAM.	Hella.	PIQA	ARC	Wino.	***Avg.***	NQ-S	NQ-M	NQ-L	Drop	FDA	SWDE	TQA	SQD	***Avg.***
M	45.3	52.7	72.1	67.6	54.9	58.5	15.8	13.0	10.6	16.9	4.1	14.8	50.8	23.6	18.7
SF	47.2	49.8	69.5	65.9	53.4	57.1	18.0	16.0	10.7	19.1	10.3	29.0	52.0	24.9	22.5
Sequential Hybrid															
MS	48.9	48.8	69.9	65.3	54.9	57.6	17.9	15.4	10.3	19.3	45.9	26.4	53.8	23.1	26.5
MFSF	52.9	52.6	71.9	68.5	55.4	60.3	17.6	15.6	10.9	20.0	46.2	38.6	53.7	24.7	28.4
Parallel Hybrid															
MF|SF (Avg)	53.5	51.9	71.4	64.1	56.1	59.4	19.1	16.0	12.4	18.6	47.8	35.8	53.3	24.6	28.5
MF|SF (MergeAttn)	54.4	53.7	71.7	68.0	57.4	**61.0**	17.9	16.9	11.8	19.4	48.4	39.7	51.7	26.0	**29.0**

Table 2: Model performance at the 1.3B scale. Due to computational constraints, we evaluate only those configurations that performed well at the 430M scale. Model Type: M = Mamba, S = SWA, F = FF layer. The order reflects the design sequence within each block. In parallel hybrids, "|" denotes parallel branches (e.g., M|SF means Mamba on one side, SWA+FF on the other). Tasks: LAM. = LAMBADA-OpenAI, Hella. = HellaSwag, ARC = ARC-Easy, Wino. = Winogrande, NQ-S = NQ-512, NQ-M = NQ-1024, NQ-L = NQ-2048, TQA = TriviaQA, SQD = SQuAD. Bold indicates the highest average performance. In both cases, the best models use hybrid architectures with merge-attention.

D.7 Similarity between SWA and Mamba Output Embeddings in Hybrid Models

We observe that sequential hybrids exhibit high similarity between SWA and Mamba outputs, especially in the larger 1.3B model, while parallel hybrids show much lower similarity, particularly in early and middle layers. This difference arises from the design: sequential hybrids pass outputs from one component to the next, naturally aligning their embedding distributions. In contrast, parallel hybrids process the same input independently, with their outputs aggregated later, leading to more distinct representations.

This structural difference impacts performance. Sequential hybrids maintain a consistent representational space, enabling stable training and strong results on tasks requiring commonsense reasoning or handling shorter contexts. However, they struggle with longer-context tasks that require richer representations. Parallel hybrids, while more sensitive to aggregation strategies due to the divergence in output spaces, can achieve better performance on complex tasks when trained effectively by leveraging the complementary strengths of both components.

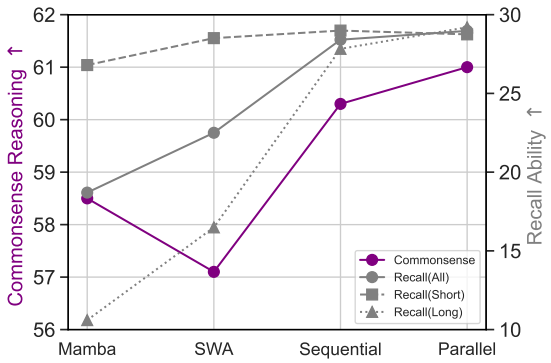


Figure 15: Performance of best performing 1.3B scale models from each architecture in commonsense reasoning and recall ability, where divided by length of context.

D.8 Identifying Critical Components in Hybrid Blocks

Figure 6 presents the performance reduction rates (y-axis) for commonsense tasks (left) and recall tasks (right) as a function of the block removed (x-axis, represented as the percentage depth from the final block). Across most configurations, the removal of the first block results in the highest performance degradation, indicating that early blocks are typically the most critical. This trend is particularly pronounced in recall tasks, where removing the first block often leads to performance drops of around 90%.

To better understand the importance of components within each block, we analyze how the removal of specific subcomponents affects performance across architectures. In sequential architectures, the first subcomponent in each block plays the most important role. In contrast, in parallel architectures, the aggregation mechanism rather than individual components like Mamba or SWA, is the most impactful. In more details, in sequential architectures, such as MFSF, where the Mamba layer is placed first, removing this initial layer leads to significant degradation, while removing the SWA layer has a milder effect. Conversely, in SFMF, which places the SWA layer first, the most substantial drop occurs when the SWA layer is removed, with the Mamba layer being less impactful (Figure 14). These results suggest that the position of the layer (i.e., being the first) has a greater influence on performance than the specific type of layer (Mamba vs. SWA). For parallel architectures, the impact of removing individual Mamba or SWA layers is less severe. Instead, the greatest degradation occurs when aggregation mechanisms such as merge-attention or projection layers are changed to a simple average.

The findings can also be related to the distribution shift caused by each component. Sequential architectures exhibit the strongest distributional shift in the first component, making it consistently important regardless of whether it is Mamba or SWA component. After this initial transformation, subsequent components tend to collapse into similar distributions, reducing their relative impact. In contrast, in parallel architectures, both the Mamba and SWA components process the same input independently. As a result, the distributional shifts introduced by each path are less pronounced, and the model can still form a reasonable representation of the input even if one component is removed. However, the aggregation mechanism causes the largest distributional shift in parallel architectures. Replacing it with simpler methods, such as averaging, can distort the combined representation from the two components, resulting in significant performance degradation.

D.9 Calculating average attending weights

We calculate Mamba hidden attention maps following Ben-Kish et al. (2024). The average attending weight is calculated with a randomly selected 100 samples of the validation set of Slimpajama of a 4k chunk. We average over all tokens and all layers.

E Dataset Strategies to Enhance Recall

E.1 Filtering the Paraphrased Dataset

We apply a filtering process to the paraphrased dataset based on the following criteria: (1) the model fails to generate a valid question and answer pair, (2) the generated answer is not present in the corresponding paragraph, or (3) the model fails to convert the example into a cloze-style task, such as when the answer does not appear at the end of the sentence. Instances that do not meet these criteria are discarded, and the processing continues with the remaining examples. For all experiments, we maintained approximately 3k training instances in the training dataset to ensure a fair comparison.

E.2 Introducing Paraphrased Data: Early vs. Late

We investigate the impact of introducing paraphrased datasets at different stages of pretraining. When added early, performance deteriorates: al-

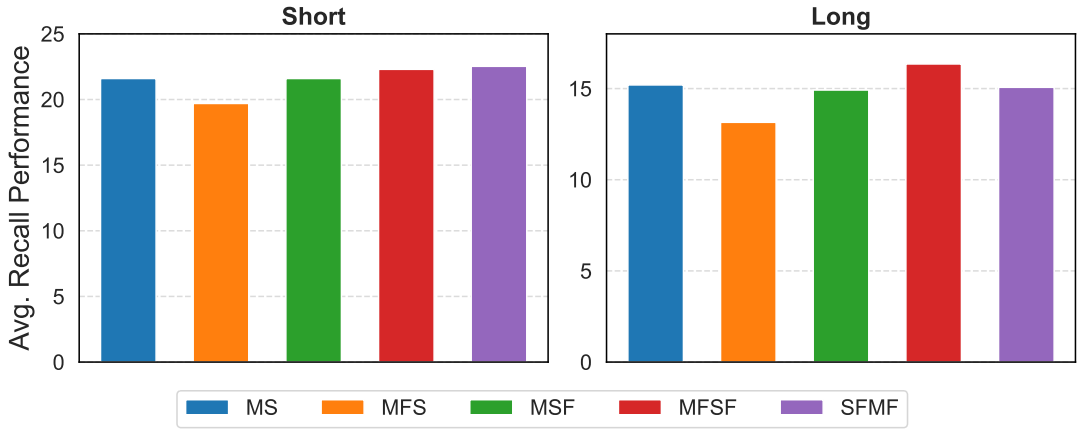


Figure 16: Comparison of average recall performance across short and long input contexts for sequential hybrids

Training Dataset	Commonsense	Recall
Original	51.5	19.1
Based (SQuAD)	45.3	22.3
NIAH	50.4	21.6
UltraChat	47.0	20.6
Ours	51.2	20.4

Table 3: Performance on commonsense reasoning and recall ability after training on the datasets listed in the "Training Dataset" column.

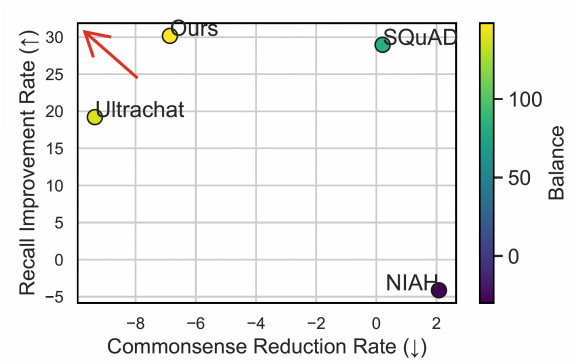


Figure 17: The upper-left region (indicated by the red arrow) represents the optimal balance between recall improvement and commonsense degradation. When training a pretrained 430M Mamba across various datasets, models trained on our dataset (paraphrased SlimPajama) consistently achieve the best balance compared to when training on other datasets.

though training loss decreases steadily, validation loss increases, suggesting overfitting. We hypothesize this is due to the model's limited language modeling ability in the early stages, making it more sensitive to data quality. Additionally, deduplication plays a critical role in preventing overfitting. In contrast, introducing paraphrased data later in continual training stage, as the model is stable, we observe that it consistently improves the recall performance.

E.3 Our Dataset Achieves the Best Balance Across Various Training Datasets

Table 5 shows the performance on commonsense reasoning and recall ability after training on datasets on SQuAD dataset from Based, NIAH, UltraChat, and Ours (paraphrased slimpajama dataset). Please note that we remove the SQuAD dataset when averaging recall ability.

E.4 Ours Also Shows Good Balance on Mamba-Only Models

Figure 17 illustrates that models trained on our dataset (paraphrased SlimPajama) tend to achieve an optimal balance, compared to those trained on alternative datasets such as NIAH, Based, or UltraChat, when finetuned on top of the pretrained 430M Mamba model.

Along with hybrid models, we observe notable improvements in recall ability with minimal or no degradation in commonsense reasoning or language modeling performance when training non-hybrid models (models using only Mamba or only SWA layers) with a dataset of 4k sequence length and 40 million tokens. The mamba-only model showed a recall improvement rate of 29.5%, whereas the SWA-only model showed a more modest improvement of 17.7%. This suggests that the Mamba-only model, despite initially exhibiting weak recall performance due to underdeveloped recall capabilities during pretraining, has significant potential for recall when further trained. Prior to our additional training, the SWA-only model outperformed the Mamba-only model in recall (SWA:

13.8, Mamba: 12.5). However, after training, the Mamba-only model learned to better retain and recall information, resulting in a recall performance of 17.7, surpassing that of the SWA-only model (16.8). Furthermore, this improvement in recall did not come at the cost of commonsense reasoning. The Mamba-only model shows a 6.87% increase in commonsense reasoning, whereas the SWA-only model shows a 2.96% decline in commonsense reasoning ability. These results suggest that our method not only benefits hybrid models but also improves the performance of various model architectures, particularly those utilizing Mamba layers.

E.5 Comparison with DeciMamba

We evaluate performance on the LongBench dataset to compare our training approach with DeciMamba, using the same base model (Table 6). Model trained with our dataset consistently yields stronger results, particularly on QA datasets, with an average improvement of +8.1 points.

E.6 Generalization Across Different Mamba-2.8B Variants

Table 4 presents the performance of various base models trained using our paraphrased dataset. To ensure a fair comparison, we evaluate three variants of the Mamba-2.8B model: Mamba-2.8B, Mamba-2.8B-Ultrachat, and Mamba-2.8B-Zephyr. Our results show consistent improvements in both commonsense reasoning and recall performance when using the paraphrased dataset. Notably, the gains are most pronounced when using the original Mamba-2.8B model as the base, suggesting that models with fewer prior instruction-tuning steps may benefit more from paraphrased augmentation.

E.7 Length of Training Dataset

As shown in Figure 18, for both sequential and parallel architectures and various model sizes, **continually training with longer chunk size result in lower reduction rate on commonsense tasks and higher improvements on recall tasks**.

Upon closer inspection (Table 5), shorter chunk sizes (e.g., 2k) significantly boost performance on short-context recall tasks but lead to notable degradation on long-context tasks. This effect is particularly pronounced in parallel models. We hypothesize that this is because, as shown in Section D.4, parallel hybrid retains layer-wise characteristics more strongly than sequential models. Additionally, the gap in performance is more substantial for recall tasks (range of around -1% to 7%) than for commonsense tasks (range of around -1% to 3%).

E.8 Number of training dataset

Figure 20 shows the reduction rate of commonsense performance (left) and the improvement rate of recall performance (right) by the number of training token (x-axis), trained with a chunk size of 4k. As the training data size increases, we observe a general improvement in both commonsense and recall performance with convergence of around 80M to 100M tokens.

F CheckList

F.1 Potential Risks

Although our experiments are conducted on publicly available datasets, we do not apply additional data cleaning. As a result, the pretrained model may produce unexpected or unintended outputs due to noise or biases present in the data.

F.2 LLM Usage

We used the free version of ChatGPT-4o to assist with grammar checking during the writing of this paper.

		Commonsense Reasoning						Recall Ability							
Base Model	Type	LAM.	Hella.	PIQA	ARC	Wino.	***Avg.***	NQ-S	NQ-M	NQ-L	Drop	FDA	SWDE	TQA	***Avg.***
Mamba		69.1	49.5	75.3	64.1	63.2	63.7	31.0	28.1	21.7	20.9	29.6	41.0	64.6	33.8
	+ Ours	67.0	64.8	76.0	68.3	62.4	65.4	41.0	37.3	27.5	31.5	32.2	41.0	71.4	40.3
Mamba-U		67.0	70.5	78.6	65.9	65.2	65.6	36.3	35.0	27.7	25.7	34.3	50.1	70.5	39.9
	+ Ours	65.9	69.7	78.3	69.8	64.0	66.9	42.6	39.4	30.8	30.8	33.6	52.4	74.8	43.5
Mamba-Z		67.9	71.2	78.4	66.2	65.0	65.6	36.8	35.1	27.8	26.1	32.8	51.6	70.4	40.1
	+ Ours	66.8	69.7	77.8	70.2	67.8	69.0	42.4	38.5	30.6	31.3	34.2	34.9	74.3	40.9

Table 4: Performance of Mamba-2.8B when continually trained on our paraphrased dataset, evaluated across different base model variants. We observe consistent improvements in both commonsense reasoning and recall capabilities, with gains more pronounced for stronger base models (e.g., Mamba). "Mamba-U" and "Mamba-Z" refer to Mamba-2.8B-UltraChat and Mamba-2.8B-Zephyr, respectively.

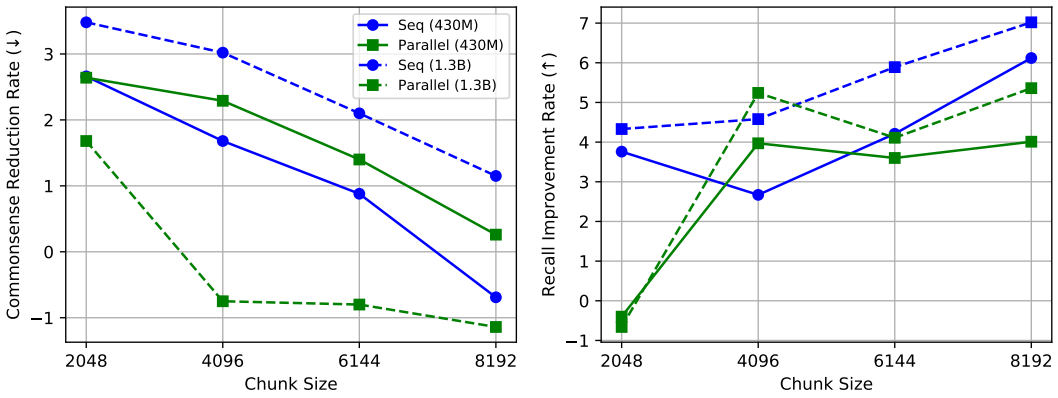


Figure 18: Commonsense reasoning reduction rate(left) and recall improvement rate(right) by changing the chunk size of the training dataset (x-axis).

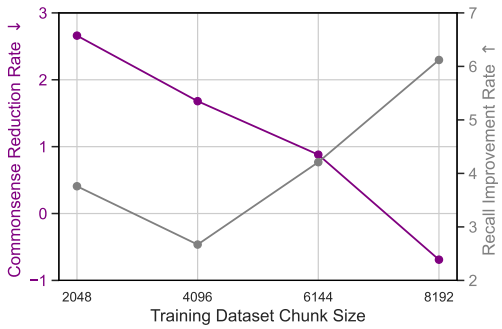


Figure 19: Commonsense reasoning reduction rate and recall improvement rate by changing the chunk size of the training dataset (x-axis).

Length	Commonsense	Recall (Short)	Recall (Long)	Recall (All)
Original	51.5	22.33	16.35	19.34
2k	50.1	24.3	16.4	19.8
4k	50.6	23.6	16.5	19.6
6k	51.0	23.7	17.0	19.9
8k	51.9	24.4	17.3	20.3

Table 5: Average commonsense and recall performance for short and long contexts as training dataset length (Length Column) varies in sequential hybrid (MFSF) training.

Benchmark	**Avg Len**	Mamba	DeciMamba	Mamba + Ours
2wikimqa	4887	3.9	**9.1**	8.0
Hotpotqa	9151	1.5	4.5	**12.5**
Musique	11214	0.9	1.7	**2.3**
Narrative QA	18409	0.9	1.7	**3.5**
Qasper	3619	5.97	**8.9**	8.5
Multifield QA	4559	11.2	18.6	**19.3**
GovReport	8734	9.8	14.9	**15.2**
QMSum	10614	**8.2**	7.1	7.3
MultiNews	2113	23.2	**24.6**	23.7
TriviaQA	8209	3.9	12.6	**36.0**
SAMSum	6258	**8.6**	7.3	6.9
TREC	5177	0.5	0.5	**27.0**
LCC	1235	8.1	8.7	**8.9**
RepoBench-p	4206	7.2	**11.0**	10.7
Passage Count	11141	0.0	**0.5**	0.0
Passage Ret. en	9289	0.0	1.5	**1.9**

Table 6: Performance over LongBench. Results of DeciMamba are from the paper (Ben-Kish et al., 2024). **Mamba+Ours** is model continual trained with our paraphrased dataset on the same base model (instruction-tuned Mamba-2.8b model). Ours tend to show high performance, especially on QA datasets.

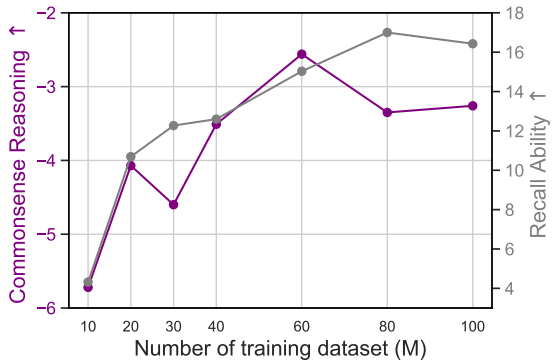


Figure 20: Commonsense reasoning and recall ability when changing the number of training dataset (x-axis).

Findings of the Third BabyLM Challenge: Accelerating Language Modeling Research with Cognitively Plausible Data

BabyLM Team

Lucas Charpentier[1], **Leshem Choshen**[2,3], **Ryan Cotterell**[4], **Mustafa Omer Gul**[5], **Michael Y. Hu**[6], **Jing Liu**[7], **Jaap Jumelet**[8], **Tal Linzen**[6], **Aaron Mueller**[9], **Candace Ross**[10], **Raj Sanjay Shah**[11], **Alex Warstadt**[12], **Ethan Gotlieb Wilcox**[13], **Adina Williams**[10]

[1]LTG, University of Oslo [2]IBM Research [3]MIT [4]ETH Zürich [5]Cornell University [6]NYU [7]ENS-PSL [8]University of Groningen [9]Boston University [10]Meta AI [11]Georgia Tech [12]UC San Diego [13]Georgetown University

Abstract

This report summarizes the findings from the 3rd BabyLM Challenge. The BabyLM Challenge is a shared task aimed at closing the data-efficiency gap between human and machine language learners. This year, the challenge was held as part of an expanded BabyLM Workshop that invited paper submissions on topics relevant to the BabyLM effort, including sample-efficient pretraining and cognitive modeling for LMs. For the challenge, we kept the text-only and text-image tracks from previous years, but also introduced a new *interaction* track, where student models are allowed to learn from feedback from larger teacher models. Furthermore, we introduce a new set of evaluation tasks to assess the "human likeness" of models on a cognitive and linguistic level, limit the total amount of training compute allowed, and measure performance on intermediate checkpoints. We observe that new training objectives and architectures tend to produce the best-performing approaches, and that interaction with teacher models can yield high-quality language models. The strict-small and interaction tracks saw submissions that outperformed the baselines. We do not observe a complete correlation between training FLOPs and performance. This year's BabyLM Challenge shows that there is still room to innovate in a data-constrained setting, and that community-driven research can yield actionable insights for language modeling.

1 Introduction

Language modeling (LM) has become increasingly compute-intensive in the past decade, and is thus often cast as the preserve of tech giants. LM research is also often dismissed as irrelevant to the study of language and mind, as the number of words required to train a state-of-the-art model is orders of magnitude greater than the number of words a human would hear in their lifetime.

To advance the science of language modeling at the academic scale and create more cognitively plausible LMs, the BabyLM Challenge encourages researchers to train Large Language Models (LLMs) with the amount of language typical of human language acquisition. This paper presents and analyzes the main findings from the third iteration of the BabyLM Challenge.[1] We also present the winning submissions and some key takeaways from the BabyLM workshop, to which participants could submit papers without needing to submit to the challenge.

The objective of the BabyLM Challenge is to train a model with 100M words or fewer. For entrants wishing to work at an even smaller scale, we also organized a 10M-word track. The challenge explicitly refrains from restrictions on anything other than the word count. In doing so, we hope to encourage new approaches that improve LLMs' sample efficiency and reveal why standard LLMs are so data hungry. Previous BabyLM iterations (Hu et al., 2024) included a multi-modal text-image track. While we keep this track, this year, we also introduced an interactive track. The interactive track enables training on direct interaction with a teacher model via the student model's generated outputs, rather than passive exposure to human-generated texts. Inspired by interactions

[1]For findings from previous years, see Warstadt et al. (2023); Hu et al. (2024). For a write-up focused on implications for psycholinguists, see Wilcox et al. (2025).

Proceedings of the First BabyLM Workshop, pages 411–432
November 8, 2025

during human language acquisition, we hoped to encourage researchers to investigate the benefits of adapting the text seen to the model's needs (see §3.3).

Summary of takeaways. As in the previous two iterations of the BabyLM Challenge, curriculum learning was a common approach. However, the most effective approaches were those that proposed architectural innovations or modifications to the training objective or procedure. Winners included a diffusion language model (Kosmopoulou et al., 2025), a mixture-of-experts model (Tapaninaho, 2025), and a reinforcement learning-based interactive approach (Martins et al., 2025).

2 Competition Details

Track Overview. The third BabyLM Challenge included four competition tracks: the returning *Strict*, *Strict-Small*, and *Multimodal* tracks and the newly added *Interaction* track.

The *Strict* and *Strict-Small* tracks require submissions to be trained on datasets of 100M and 10M words or less, respectively. Participants were free to use the provided BabyLM corpus or construct their own training datasets, provided that they adhered to the track's word limitations. Models in this track were evaluated on language-only evaluation tasks.

In the *Multimodal* track, participants trained multimodal vision-language models. Participants were allowed to use any model and training procedure, provided that the model could assign (pseudo) log-likelihoods to strings of text, conditioned on input images. Submissions could be trained on any arbitrary dataset of 100M words or less, including our provided corpus for the *Multimodal* track, which is split evenly between text-only and paired image-text data. Models in this track were evaluated on both language-only and multimodal tasks.

New to this year, the *Interaction* track enabled participants to explore how feedback and interaction could assist with sample-efficient modeling. Here, an external model different from the participants' submission model could be incorporated into the training pipeline. Participants were prohibited from exposing the external model's weights, hidden states, or output distribution to the submission model, but were otherwise unrestricted in how they instantiated "interactions." The external model could, for instance, give scalar or natural language feedback to the submission model or produce training data conditioned on the submission model's outputs. Similar to previous tracks, the submission model could be exposed to at most 100M external words, which could come either from regular datasets or the external model. Furthermore, the submission model could not generate more than 100M words of its own. Finally, we restricted the external model to a pre-determined list of models (namely Llama3.1-8B-Instruct, Llama3.2-3B-Instruct, Llama3.1-1B-Instruct (Dubey et al., 2024), and any language model below 1B parameters). Participants were allowed to fine-tune these models without any restriction. Models in this track were evaluated on language-only evaluation tasks.

The data composition of the corpora for each competition track is described in full in Table 1.

Training Duration Limitations. This year, we restricted models to a fixed amount of training data exposure, counting repeated passes over the same input, specifically at most 100M words for the *Strict-Small* track and at most 1B words for the other tracks. This decision was motivated by two goals of BabyLM. Firstly, BabyLM aims towards developmentally plausible training. While memories of inputs could have an impact on learning beyond the initial exposure, dozens or hundreds of repeated exposures are developmentally implausible. Secondly, BabyLM aims towards democratizing pretraining research. We observed in the 2024 BabyLM Challenge that larger numbers of training epochs improved model performance, which gives groups with greater computational resources a significant advantage if no limitations exist. Although the new limitation does not eliminate all advantages of greater compute, such as for hyperparameter tuning, it helps ensure that successful training procedures are more reproducible and accessible to teams with modest resources.

Intermediate Checkpoints. We additionally required participants to submit intermediate model checkpoints corresponding to different word exposure amounts. We specifically ask for checkpoints for every 1M words until 10M words are seen, every 10M words until 100M words are seen, and every 100M words until 1B words are seen. Each checkpoint would then be evaluated on a subset of less compute-intensive tasks. The motivation behind this is that the training dynamics of LMs can be compared to the learning trajectories of children, which is valuable from the cognitive modeling per-

spective. Results from this analysis are presented in Figure 5.

3 Baselines

In this section, we detail the baselines and their associated training procedures for each competition track. When possible, we set winning entries from the past competition year as the baseline for a given track. Each baseline is meant to encourage participants to innovate and improve beyond existing models and approaches.

3.1 Strict and Strict-Small Tracks

For the *Strict* and *Strict-Small* tracks, we used last year's winning submission, GPT-BERT (Charpentier and Samuel, 2024), and the GPT-2 Small (Radford et al., 2019) architecture naively trained with an auto-regressive language modeling loss as baselines. GPT-BERT is based on the architecture of LTG-BERT (Samuel et al., 2023), a BERT-style model developed to work with low amounts of data. It uses disentangled attention from DeBERTa (He et al., 2021), both pre- and post-layer normalization as in NormFormer (Shleifer et al., 2021), span masking, and GEGLU activation functions in the feed-forward layers. In addition to using LTG-BERT as a base, GPT-BERT uses both the masked and auto-regressive language modeling objectives to train the models. To achieve this, the authors used a variation of standard masked language modeling called masked next token prediction, where the outputs are shifted in the same way as in the auto-regressive training. By training with both objectives, the models can be used both as an encoder and a decoder.

As GPT-BERT is trained with both masked and autoregressive language modeling losses, we train three variants for it: one focused on the autoregressive loss, another focused on the masked loss, and finally, another with equal focus on both losses. Baselines for each track were trained using the corresponding BabyLM corpus.

GPT-BERT. In line with the challenge requirements, we train the *Strict* and *Strict-Small* models for 10 epochs. Our *Strict* models have around 120M parameters with 12 layers and 12 attention heads. We use a batch size of 131 072 tokens and train for 12 330 steps. Our *Strict-Small* models have around 31M parameters with 12 layers and 6 attention heads. We use a batch size of 16 384 tokens and train for 9 914 steps. For both tracks, we use a warmup-cosine-cooldown learning rate scheduler with a maximum learning rate of 7×10^{-3}. The first 1.6% of steps are used for linear warmup, and the final 1.6% of steps are used for linear cooldown. For the masked objective, we start the masking ratio at 0.3 and linearly decay it to 0.15. We use a sequence length of 128 tokens for the first 60% of training steps, we then increase the sequence length to 256 tokens for the next 20%, and for the final 20% we use a sequence length of 512.

We train three variants of GPT-BERT. The auto-regressive focus uses a 93.75-6.25 mix of auto-regressive to masked ratio. The mixed focus uses a balanced 50-50 mix of auto-regressive to masked ratio. Finally, the masked focus uses a 6.25-93.75 mix of auto-regressive to masked ratio. All three models in each track are evaluated both in the masked next-token prediction (MNTP) and auto-regressive styles. A complete list of hyperparameters can be found in the HuggingFace Model Hub; the HuggingFace names of the models can be found in Appendix B.

GPT-2. We additionally train the GPT-2 Small (Radford et al., 2019) with a purely auto-regressive loss as a naive baseline. We first chunk the BabyLM corpus into datapoints of 512 tokens each. The model is trained for 10 epochs with a batch size of 16 (containing 8192 tokens per step). We use a learning rate of 5×10^{-5} with a cosine-decay scheduler that warms up the learning rate in the initial 1% of training. We use AdamW (Loshchilov and Hutter, 2019) as the optimizer.

3.2 Multimodal Track

As no submissions outperformed our *Multimodal* baselines in the 2024 BabyLM Challenge, we re-released them for this year. We train the GIT (Wang et al., 2022) and Flamingo (Alayrac et al., 2022) architectures on the BabyLM corpus for the *Multimodal* track, and use a frozen DINO model with the ViT-B/16 architecture as the image encoder (Caron et al., 2021).

We perform training on the BabyLM corpus for the *Multimodal* track. We train the models for 4 epochs, where each epoch consists of one pass over the text-only half of the corpus and four passes over the remaining image-text paired data (resulting in 250M word exposures per epoch). We use a learning rate of 10^{-4}, with a linear learning

rate scheduler, and train with the AdamW optimizer (Loshchilov and Hutter, 2019).

3.3 Interaction Track

For the *Interaction* track, we provide a baseline that explores how corrections in natural language can be incorporated into language model training. We split training into 20 rounds of interaction. At each round, the student model, initialized with the GPT-2 Small architecture (Radford et al., 2019), is given incomplete data points sampled from the BabyLM training corpus. For each data point, the student samples a completion. The teacher model, chosen to be Llama-3.1-8B-Instruct (Dubey et al., 2024), is then prompted to revise the student model's completion based on grammaticality, coherence, and relevance to the input. The student model is then first trained with the language modeling loss on the full teacher-corrected datapoint and is then further finetuned with SimPO (Meng et al., 2024), a preference optimization algorithm, where the teacher and student completions are the winning and losing responses, respectively.

We split each constituent dataset of the BabyLM corpus into 20 equally sized chunks prior to training. At each round, a chunk is sampled at random from each constituent dataset without replacement. Each chunk is then split into data points consisting of 512 tokens. The student is provided the first 256 tokens of each data point as context for generation. We then sample student completions with nucleus sampling (Holtzman et al., 2020) where $p = 0.8$. Teacher corrections are similarly sampled using nucleus sampling with $p = 0.8$. The prompt can be found in the Appendix.

We optimize the student model with AdamW (Loshchilov and Hutter, 2019) with a learning rate of 5×10^{-5} and set $\beta = 2$ and $\gamma = 1$ for SimPO. We add the language modeling loss on the winning completion, with a scaling coefficient of 0.2, as a regularizer during preference optimization training, following Dubey et al. (2024). For each round of interaction, we perform 7 epochs of training with the regular language modeling loss on full teacher-corrected datapoints, followed by 2 epochs with SimPO.

4 Evaluation

For evaluation, we kept the tasks from previous year's edition. For the *Strict* and *Strict-Small* these are the (Super)GLUE suite of NLP tasks (Wang et al., 2018, 2019), the linguistic minimal pairs of BLiMP (Warstadt et al., 2020), and the Elements of World Knowledge (EWoK) dataset (Ivanova et al., 2024), which measures pragmatic, commonsense, and discourse knowledge. For evaluation of the *Multimodal* track, we test again on Visual Question Answering (VQA, Agrawal et al., 2015; Goyal et al., 2017), WinoGround (Thrush et al., 2022), and DevBench (Tan et al., 2024).

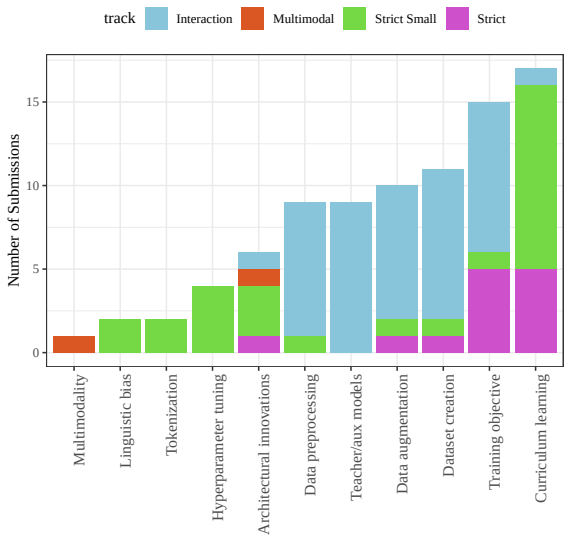


Figure 1: **Number of submissions by approach**. Curriculum learning was again the most popular approach. This year, we encouraged more teacher/auxiliary-model approaches in the interaction track.

4.1 New Tasks

This year, we additionally included tasks that measure *psychometric fit* to human language learners and linguistic abilities of aspects not covered by BLiMP. For selecting these tasks, we focused on the following two aspects of a model being 'human-like': i) connecting model behavior and internals to cognitive aspects of human language processing, such as reading time prediction, and ii) assessing how human-like a model's generalizations are on various tasks related to reasoning and morphology. We excluded tasks that could not be reasonably acquired from the BabyLM training data. Below, we describe the tasks in more detail.

Morphological Generalization Weissweiler et al. (2023) introduce a task for testing morphological generalization, based on a past tense formation task of nonce ("*wug*") words: e.g. *veed* $\rightarrow$ *ved*/*veeded*/*vode*. Similar to this task, we also include the task of Hofmann et al. (2025), in which nonce adjectives are *nominalized* as

Dataset	Description	# Words (multimodal)	# Words (strict)	# Images
Localized Narratives[a]	Image Caption	27M	–	0.6M
Conceptual Captions 3M[b]	Image Caption	23M	–	2.3M
CHILDES[c]	Child-directed speech	14.5M	29M	–
British National Corpus (BNC), dialogue portion[d]	Dialogue	4M	8M	–
Project Gutenberg (children's stories)[e]	Written English	13M	26M	–
OpenSubtitles[f]	Movie subtitles	10M	20M	–
Simple English Wikipedia[g]	Written Simple English	7.5M	15M	–
Switchboard Dialog Act Corpus[h]	Dialogue	0.5M	1M	–
Total	–	100M	100M	2.9M

Table 1: Datasets for the *Multimodal* and *Strict* tracks of the 3rd BabyLM Challenge. Word counts are approximate and subject to slight changes. [a]Pont-Tuset et al. (2020) [b]Sharma et al. (2018) [c]MacWhinney (2000) [d]Consortium (2007) [e]Gerlach and Font-Clos (2020) [f]Lison and Tiedemann (2016) [g]`https://dumps.wikimedia.org/simplewiki/` [h]Stolcke et al. (2000)

either an -ity or -ness noun: e.g. *cormasive* → *cormasiveness*/*cormasivity*. We evaluate these tasks against human predictions: from the participant responses included in each of the above papers, we derive a distribution over human-preferred inflections. Our score for this task is then a correlation between the model's probability for each inflection against the human preference distribution.

Entity Tracking Kim and Schuster (2023) tests entity state tracking in LMs, by describing a sequence of actions placing and removing items to and from various numbered boxes and evaluating a model's understanding of the contents of each box at a given moment. We revised the evaluation of this task to evaluate LMs' ability to assign the highest probability to the correct continuation (akin to BLiMP and EWoK) rather than requiring the model to generate the correct completion as in the original operationalization. This was done to enable simpler, zero-shot evaluation. We construct five candidate continuations, one of which is the ground-truth. Distractor continuations were constructed by copying prior contents of a given box, contents of an adjacent box, or the result of the most recent action. They were also synthetically generated by randomly swapping, adding, and removing objects from the box state.

Concept Knowledge Misra et al. (2023) introduce a task for testing the property knowledge of language models and whether they can infer that properties of superordinate concepts are inherited by subordinate concepts, each represented by nonce words. The dataset is composed of minimal pair sentences, and models are evaluated by whether they assign a higher probability to the correct sentence.

Reading Time Prediction de Varda et al. (2023) Connects LM predictions to human reading times, allowing us to assess to what extent LM processing is aligned with human language processing. To measure this, we do a correlation between the surprisal score (defined as the negative log probability of a word) of a word for a language model and either the time it took for a human to read the word or the time spent looking at the word. The more correlated the two metrics are, the more human-like a model is, following the previously established relationship between surprisal and reading time, wherein words that take longer to read are associated with higher surprisal scores (Wilcox et al., 2020, 2023).

Word Learning Chang and Bergen (2022) present a benchmark for tracking word surprisal across training checkpoints to extract learning curves and compute ages of acquisition for vocabulary items. We compute surprisal scores as the negative log probability of target words given their contexts in the C4-en-10k test set (a shuffled subset of the first 10,000 records from the English portion of the C4 corpus) across training steps. We then fit sigmoid functions to each word's learning trajectory. In the end, the benchmark enables direct comparison between the language model and child language development by computing correlation scores between model-derived and human Age-of-Acquisition data from the WordBank repository (Frank et al., 2016).

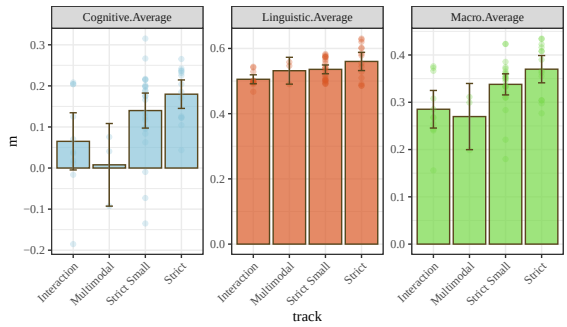


Figure 2: **Scores by track**. Despite the greater flexibility and tailored supervision allowed in the interaction track, performance was generally higher in the strict track. Multimodal models remain difficult to train, perhaps due to the track design (as discussed in Ganescu et al. (2025)).

4.2 Evaluation Pipeline

As in previous years, we distributed an open-source evaluation pipeline that could be run by all participants.[2] We rewrote the evaluation pipeline from scratch so as to make the structure of the repository significantly simpler than in previous years. This allowed participants to adapt it to their needs or unique architectures and debug any potential issues, as improving the computation efficiency. We provided a HuggingFace version that could be re-written to use only PyTorch modules.

Hidden Tasks As in previous years, we released a set of *hidden* evaluation tasks to control for overfitting to the public evaluation tasks. The hidden tasks this year were COMPS (Misra et al., 2023), the past tense formation *wug* task (Weissweiler et al., 2023), and the word learning trajectory task (Chang and Bergen, 2022). We released these tasks two weeks before the submission deadline.

Zero-shot vs. Finetuning A criticism of the evaluation procedure in previous editions was that the finetuning tasks presented a considerable computational overhead. We investigated to what extent these tasks can be evaluated using zero-shot prompting instead, but unfortunately concluded that the limited data size does not allow for robust in-context learning to emerge.[3] Therefore, we kept the existing finetuning tasks in (Super)GLUE but made the finetuning more efficient in two ways: subsampling large tasks and eliminating highly correlated tasks.

First, we sub-sampled the finetuning tasks of (Super)GLUE larger than $10,000$ training samples down to $10,000$. In our tests, we found that randomly subsampling large datasets like MNLI down to $O(1e4)$ still reliably differentiated between existing open-source models on the HuggingFace Model Hub without significantly increasing the variance due to our subsampling procedure: different subsamples of size $O(1e4)$ still gave the same stable ranking across open-source models after finetuning. Second, if models' performances on two tasks were consistently highly correlated with each other, such as with MNLI and QNLI, we eliminated one of the two tasks from our evaluations. Ultimately, we kept the following tasks from (Super)GLUE: BoolQ, MultiRC, RTE, WSC, MRPC, QQP, and MNLI. For any of these tasks larger than $10,000$ training samples, we subsampled down to $10,000$.

Next to this, we also release the evaluation tasks in two ways: a *fast* and *full* version. The *fast* evaluation consists of 20% of the data of each task (including the zero-shot tasks). This lessens the computational overhead that comes with our introduction of the evaluation of intermediate checkpoints: we only require the *full* evaluation to be run on the final model checkpoint.

5 Submission

This year, we used HuggingFace Spaces, HuggingFace Model Hub, and OpenReview for the submissions to both the workshop and challenge.

Challenge Results Submission. The participants to the challenge had to submit their results, both for the final checkpoint and intermediate checkpoints, through a leaderboard found in a HuggingFace Spaces.[4] The participants were required to submit their predictions in a JSON format; for predictions of the final model, each example consisted of an ID and a value (a text completion for non-classification tasks, and a label for classification tasks). For the intermediate checkpoints, the participants submitted the subtask scores for each checkpoint.

[2] github.com/babylm/evaluation-pipeline-2025

[3] Olsson et al. (2022) show that the *induction heads* required for in-context learning develop only after exposure to 2.5–5 billion tokens. Developing sample-efficient methods that enable such mechanisms to emerge under much smaller data budgets remains an exciting prospect for BabyLM-related research.

[4] BabyLM-community/babylm-leaderboard-2025-all-tasks

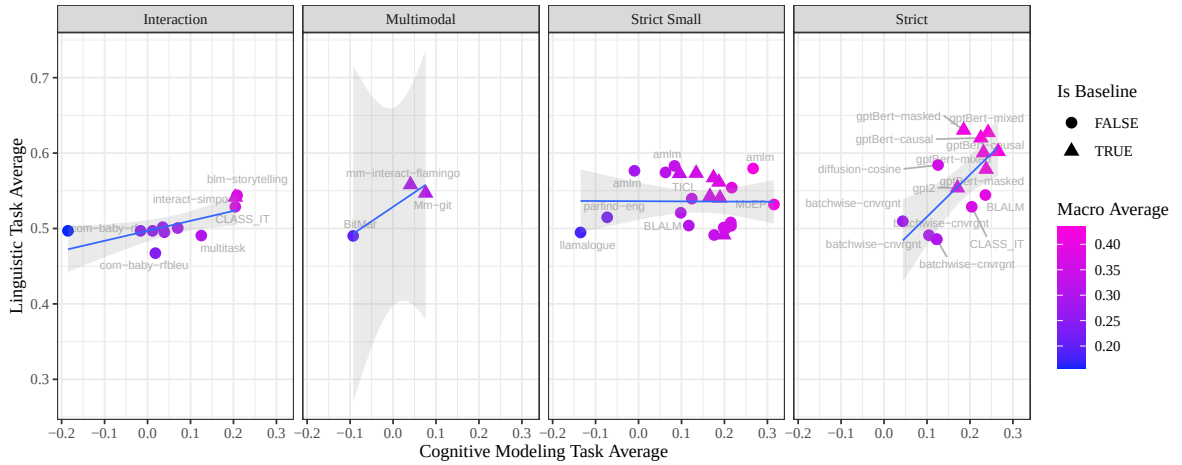


Figure 3: **Overview of the results.** We found a positive correlation between linguistic and cognitive modeling task performance, except for the *Strict-Small* track. The baselines (winning methods from previous years' challenges) remain strong, especially in the multimodal and strict tracks.

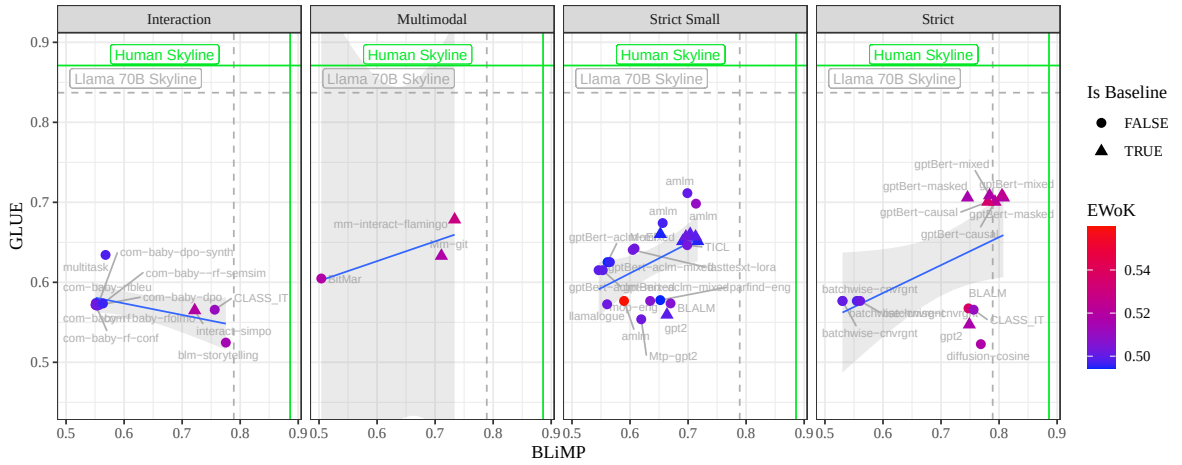


Figure 4: **Comparison of Results on BLIMP and GLUE to human scores.** Some models in the *Strict* and *Interaction* tracks are comparable to a Llama 70B parameter model on the BLiMP task. Models still fall short of skylines and human scores for GLUE.

Submission Form. In addition to submitting their results, the participants were required to fill in additional details about their training in the HuggingFace submission. These included: hyperparameters such as learning rate, scheduler, number of epochs, size of model, seed, and batch size, to name a few; information on the training dataset; number of FLOPS for both training and development; preprocessing or augmentation of data; and a short description of their model. The form can be found on the submit tab of the leaderboard.

Paper Submission. The participants were asked to submit their papers through OpenReview. Challenge participants were asked to submit papers detailing their methodology, research, and findings. Those participating in the associated BabyLM workshop were asked to submit papers thematically related to the goals of the challenge.

Artifact Submission. The participants of the challenge were also required to make their models and intermediate checkpoints available by submitting them to the HuggingFace Model Hub.

6 Competition Results

In this section, we describe the results of the competition, track winners and our selections for Outstanding Papers, which were chosen from both the challenge and workshop paper submissions. We received 32 papers to the workshop, 12 papers to the challenge, and 32 models to the challenge leaderboard. The submission counts per track are in Table 2. Similar to last year, we found low participa-

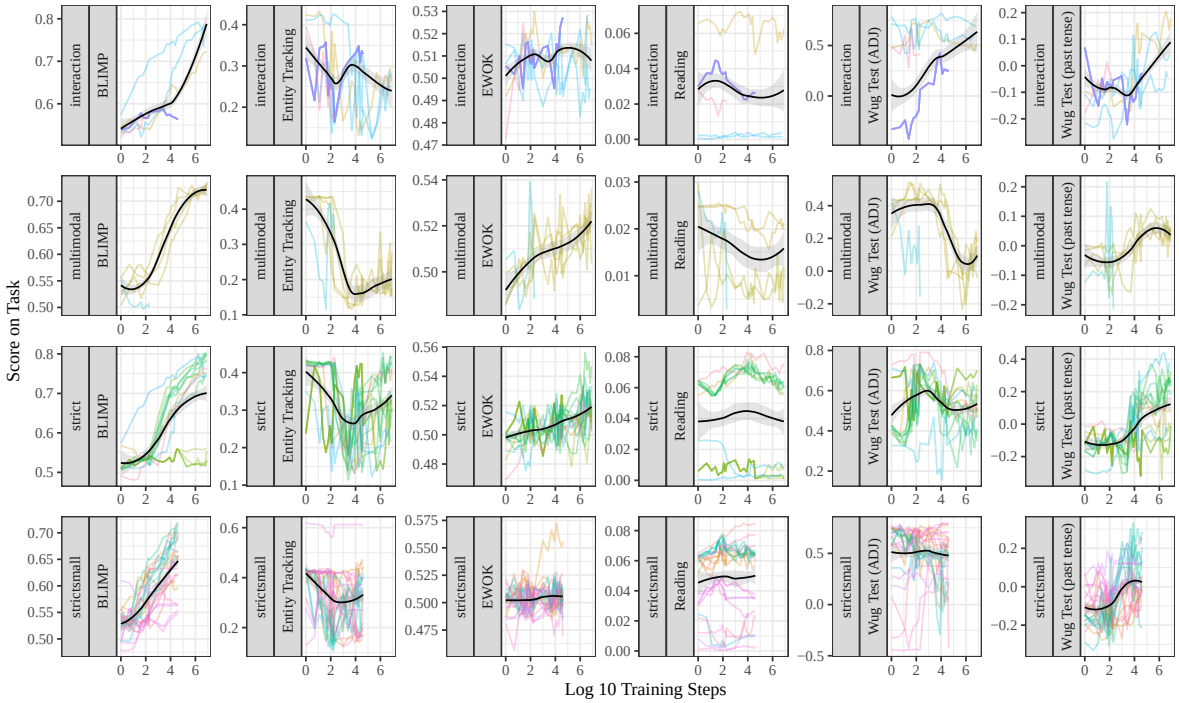


Figure 5: **Model Performance on Tasks over Training** Colors show scores for individual models; black lines show averages. Models generally show improvement for BLiMP and EWOK, while scores for reading-time predictions wug tests and entity tracking are more variable.

Track	# Models	# Participants
Strict-Small	15	9
Strict	7	4
Multimodal	1	1
Interaction	10	4
Total	*32*	*15*

Table 2: Total number of models and participants per track. This includes both participants in the challenge and workshop. Participants who submitted to multiple tracks are counted once in the total.

tion in the *Multimodal* track and received only one submission.

The breakdown of participants by affiliation and home country is as follows (submissions with multiple affiliations/countries are counted more than once): Germany (7), United States (6), England (5), Italy (3), Philippines (2), Switzerland (2), Denmark (2), Netherlands (2), Scotland (2), Japan (2), Sweden (2), Austria (2), Czechia (2), Turkey (1), India (1), Israel (1), Taiwan (1), Romania (1), Australia (1), Slovakia (1), South Korea (1), Ethiopia (1), Poland (1), Greece (1), Finland (1), Canada (1).

6.1 Winning Submissions

Human-likeness metrics were considered separate from accuracy metrics, such that a system could win either with respect to NLP task performance *or* human-likeness. We gave separate awards for both metrics.

***Strict* Track.** The winner of the human-likeness metric is CLASS-IT by Capone et al. (2025), which proposes to fine-tune small-scale LMs on a general instruction-following dataset. For the NLP tasks, the Simple Diffusion model by Kosmopoulou et al. (2025) is the winner; this is a diffusion *masked* language model.

***Strict-Small* Track.** The winner of the human-likeness metric is MoEP by Tapaninaho (2025), which employs modular mixtures of experts; this method achieves particularly high scores in the AoA task. For the NLP tasks, the AMLM-Hard-Decay model by Edman and Fraser (2025) is the winner; this method entails dynamically choosing which tokens in an input sequence to mask based on which are most difficult to predict according to the model.

***Interaction* Track.** For the *Interaction* track, we have a single winner, BLM by Martins et al. (2025),

Model	Human-likeness	NLP score	Macro Average	Vision Average
STRICT				
Best Models				
CLASS-IT*	20.4	52.9	36.6	—
Simple-Diffusion†	12.6	58.4	35.5	—
Batchwise-convergent$_{\text{main}}$	12.3	48.6	30.4	—
BLaLM‡	23.6	54.4	39.0	—
Baselines				
GPT-BERT-causal$_{\text{MNTP}}$	22.5	62.0	42.3	—
GPT-BERT-causal$_{\text{AR}}$	**26.5**	60.2	43.4	—
GPT-BERT-mixed$_{\text{MNTP}}$	24.2	62.7	**43.5**	—
GPT-BERT-mixed$_{\text{AR}}$	23.2	60.1	41.6	—
GPT-BERT-masked$_{\text{MNTP}}$	18.5	**63.0**	40.8	—
GPT-BERT-masked$_{\text{AR}}$	23.8	57.8	40.8	—
GPT2	17.1	55.4	36.2	—
STRICT-SMALL				
Best Models				
MoEP*	**31.5**	53.2	**42.3**	—
AMLM-Hard-Decay†	8.4	**58.3**	33.3	—
GPT-BERT$_{\text{ACLM-6k-MNTP}}$	21.5	50.3	35.9	—
AMLM-Hard	26.7	58.0	**42.3**	—
Baselines				
GPT-BERT-causal$_{\text{MNTP}}$	17.4	56.7	37.1	—
GPT-BERT-causal$_{\text{AR}}$	18.7	56.1	37.4	—
GPT-BERT-mixed$_{\text{MNTP}}$	13.4	57.3	35.4	—
GPT-BERT-mixed$_{\text{AR}}$	16.6	54.3	35.4	—
GPT-BERT-masked$_{\text{MNTP}}$	9.5	57.3	33.4	—
GPT-BERT-masked$_{\text{AR}}$	18.9	54.0	36.5	—
GPT2	19.8	49.1	34.5	—
MULTIMODAL				
Best Models				
BitMar‡	-9.3	49.0	19.9	26.7
Baselines				
Flamingo	4.1	**55.8**	29.9	49.3
Git	**7.6**	54.7	**31.1**	**49.7**
INTERACTION				
Best Models				
BLM*†	**20.8**	**54.4**	**37.6**	—
CLASS-IT	20.4	52.9	36.6	—
llamalogue$_{\text{rfOLMo-score}}$	7.0	50.1	28.5	—
Baselines				
SimPO	20.4	54.1	37.3	—

Table 3: Human-likeness, NLP task, macro average, and vision scores for the best models and baselines per track for the challenge. Boldened results represent the best score per track. * are the track winners for the human-likeness score. † are the track winners for the NLP task score. ‡ are workshop papers, while other models are from the challenge.

who achieved the highest score in both the human-likeness and NLP task metrics. BLM employs Llama as an interactive teacher model; the student generates a completion to a story, and the teacher scores the generated completion based on coherence, readability, and creativity. These scores are propagated as training signals to the student via a reinforcement learning-based approach.

6.2 Outstanding Paper Awards

In addition to the BabyLM Challenge winners, we gave 3 outstanding paper awards to papers that were especially interesting and likely to have significant impact for those in the community. We considered papers from both the BabyLM Challenge and BabyLM Workshop.

Are BabyLMs Deaf to Gricean Maxims? A Pragmatic Evaluation of Data-Limited Language Models. (Askari et al., 2025) This paper introduces a benchmark for evaluating the sensitivity of cognitively plausible language models to Gricean maxims. Using maxim-adhering and maxim-violating examples, it is found that pragmatic abilities improve with scale,but also that models trained on 100M words fall well short of children's abilities. Reviewers appreciated that this work contributed to an underexplored evaluation dimension, the analyses, and the solid grounding in relevant literatures.

Looking to Learn: Token-wise Dynamic Gating for Low-Resource Vision-Language Modelling. (Ganescu et al., 2025) This paper analyzes how best to make use of multimodal (text-image) data when training on cognitively plausible text corpora. The authors explore token-wise gating, channel attention, and auxiliary contrastive training objectives; these yield multimodal models that outperform the baselines. This paper features thorough analyses and strong results, and also discusses ways in which the constraints of the BabyLM Challenge indirectly limit the performance of multimodal models.

Teacher Demonstrations in a BabyLM's Zone of Proximal Development for Contingent Multi-Turn Interaction. (Salhan et al., 2025a) This paper introduces ContingentChat, a framework for evaluating and improving the *contingency*, i.e., the relevance and meaningfulness of multi-turn dialogues between student and teacher models. The authors introduce a post-training pipeline based on the Switchboard corpus and a teacher model; the method improves the grammaticality and cohesiveness of small-scale language models' generations. Reviewers appreciated the strong grounding in the developmental psychology literature.

7 Discussion

High-level takeaways. While curriculum learning remains popular, the best-performing approaches were again based on modifications to the pretraining objective or the model architecture. Diffusion MLMs, reinforcement learning with a teacher model, and mixture-of-experts approaches were especially effective. Relatedly, we notice that model performance is not necessarily tied to the total amount of compute. The relationship between these two is plotted in Figure 6 and we only find a positive correlation for the *Interaction* track. As in previous challenges, surprisingly simple approaches like better data preprocessing or hyperparameter tuning also showed performance gains over simple baselines. Now that we have included human-likeness evaluations, we can more confidently state that these methods are effective not just for improving performance on NLP tasks, but also for building better cognitive models of language processing.

Training dynamics. Visualization of training dynamics is given in Figure 5. For all models, BLiMP performance increases with the number of pretraining words. WUG past-tense performance also scales with pre-training words, but far less monotonically: there is no change in performance for the first 10–50M words. Afterwards, a phase shift occurs, and WUG performance begins to increase more monotonically with the number of words in the training corpus. Perhaps this reflects a movement from overgeneralization or memorization toward true generalization; further analyses in this low-data setting would be interesting. Entity tracking shows what appears to be U-shaped scaling for *Strict* and to a lesser extent *Strict-Small* models (Wei et al., 2023), where performance starts high, drops, and then increases again. Other tasks like reading time prediction and WUG adjective performance do not demonstrate a strong relationship with number of pretraining words.

Planned changes to future challenges. Ganescu et al. (2025) points out ways in which the provided vision embeddings for the multimodal track may constrain performance. Indeed, working with

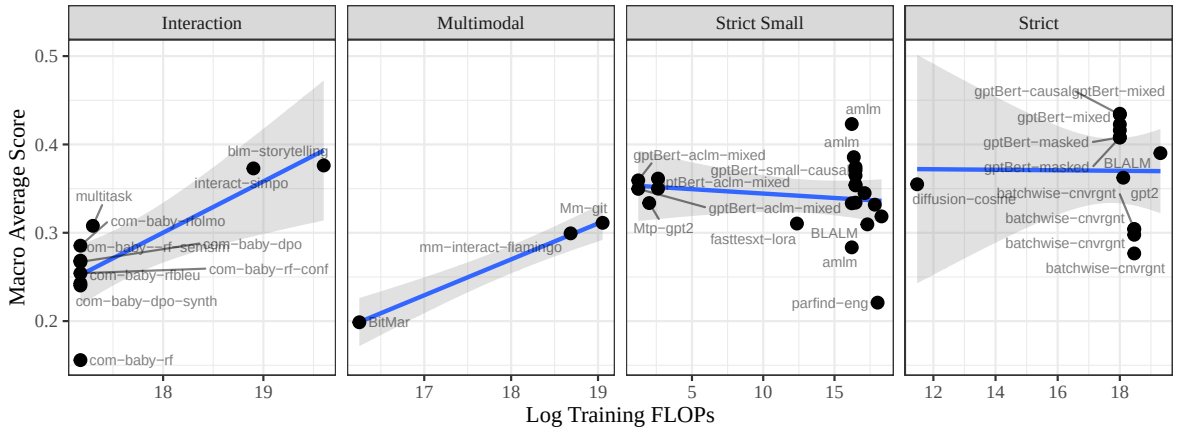


Figure 6: **Average score by flops used in training** We do not observe a strong relationship between the amount of compute used and the performance of the resulting model in the *Strict-Small* and *Strict* tracks.

these embeddings may be less straightforward than simply training the language and vision parts of a vision-language model from scratch. To encourage better performance and greater numbers of multimodal submissions in future challenges, we will consider moving to a dataset with a more open copyright license, such that participants will be able to train their own end-to-end models without needing to go through the current circumspect data download process.

8 Conclusion

The Third BabyLM Challenge shows that significant progress can be made in language modeling with academic-scale compute. With 32 models submitted from 26 countries, the challenge revealed several insights. Some agreed with the findings of previous years, for example, that training objective and architectural modifications were particularly effective. Some findings were novel this year, for example, that effective interactive approaches could be deployed using open-source teacher models. We also observed that the relationship between training FLOPs and performance was not nearly as strong this year as it was last year. Our controlled setup reveals that some approaches can outperform others based on methodological qualities distinct from how much compute they allow us to use.

Looking forward, we envision BabyLM continuing to evolve in its scope and focus. We hope to modify the multimodal evaluations to encourage more flexibility in future years. We also hope to continue exploring the value of tailored supervision and reinforcement learning-based approaches, as encouraged in the interaction track. While it is not currently as effective as simply pretraining on natural language corpora, we believe that this will continue to be a method of interest in both large- and small-scale language modeling research. By broadening our focus to include more language modeling methods, and by controlling for compute this year, we aim to inspire novel approaches that truly innovate beyond simply enabling greater compute to be spent. The strong participation and results this year suggest that the BabyLM community is well-positioned to pursue these ambitious goals, and ultimately to continue iterating towards the goal of human-like sample efficiency in language learning.

Acknowledgments

We are grateful to the BabyLM Challenge participants for making this challenge a consistent success. Our findings would not be nearly as interesting without their ambition and creativity, and their feedback on the logistics of the challenge itself, including the evaluation pipeline and training data. We are also grateful to the organizers of EMNLP for their efforts in hosting BabyLM this year.

Author Contributions

Primary Organizers Lu.Cha., Le.Cho., M.O.G., M.Y.H., J.L., J.J., A.M., C.R., E.G.W., Ad.Wi.

Pipeline implementation Lu.Cha., M.O.G., J.L., J.J.

Baseline model training Lu.Cha., M.O.G., J.L.

Communications with participants Le.Cho., E.G.W., M.Y.H

Training dataset compilation Al.Wa.

Reviewing submissions Lu.Cha., Le.Cho., M.Y.H., J.J., A.M., C.R., Al.Wa., E.G.W., Ad.Wi.

Initial draft on findings paper Lu.Cha., Le.Cho., M.O.G., M.Y.H., J.J., A.M., C.R., E.G.W., R.S.S.

Editing R.C., T.L., Ad.Wi., E.G.W., R.S.S.

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *The 36th Conference on Neural Information Processing Systems*.

EUHID AMAN, Esteban Carlin, Hsing-Kuo Kenneth Pao, Giovanni Beltrame, Ghaluh Indah Permata Sari, and Yie-Tarng Chen. 2025. BitMar: Low-Bit Multimodal Fusion with Episodic Memory for Edge Devices. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Raha Askari, Sina Zarrieß, Özge Alacam, and Judith Sieker. 2025. Are BabyLMs Deaf to Gricean Maxims? A Pragmatic Evaluation of Sample-efficient Language Models. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Ansar Aynetdinov and Alan Akbik. 2025. Babies Learn to Look Ahead: Multi-Token Prediction in Small LMs. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Necva Bölücü and Burcu Can. 2025. A Morpheme-Aware Child-Inspired Language Model. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Luca Capone, Alessandro Bondielli, and Alessandro Lenci. 2025. CLASS-IT: Conversational and Lecture-Aligned Small-Scale Instruction Tuning for BabyLMs. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

BNC Consortium. 2007. *The British National Corpus, XML Edition*. Oxford Text Archive.

Andrea de Varda, Marco Marelli, and Simona Amenta. 2023. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Lukas Edman and Alexander Fraser. 2025. Mask and You Shall Receive: Optimizing Masked Language Modeling For Pretraining BabyLMs. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Olivia La Fiandra, Nathalie Fernandez Echeverri, Patrick Shafto, and Naomi H. Feldman. 2025. Large Language Models and Children Have Different Learning Trajectories in Determiner Acquisition. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Michael C. Frank, Mika Braginsky, Daniel Yurovsky, and Virginia A. Marchman. 2016. Wordbank: an open repository for developmental vocabulary data*. *Journal of Child Language*, 44:677 – 694.

Achille Fusco, Maria Letizia Piccini Bianchessi, Tommaso Sgrizzi, Asya Zanollo, and Cristiano Chesi. 2025. Linguistic Units as Tokens: Intrinsic and Extrinsic Evaluation with BabyLM. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference*

on Empirical Methods in Natural Language Processing.

Eleni Fysikoudi, Sharid Loáiciga, and Asad B. Sayeed. 2025. Active Curriculum Language Modeling over a Hybrid Pre-training Method. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Bianca-Mihaela Ganescu, Suchir Salhan, Andrew Caines, and Paula Buttery. 2025. Looking to Learn: Token-wise Dynamic Gating for Low-Resource Vision-Language Modelling. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Yuan Gao, Suchir Salhan, Andrew Caines, Paula Buttery, and Weiwei Sun. 2025. BLiSS: Evaluating Bilingual Learner Competence in Second Language Small Language Models. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Anita Gelboim and Elior Sulem. 2025. TafBERTa: Learning Grammatical Rules from Small-Scale Language Acquisition Data in Hebrew. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1).

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Patrick Haller, Jonas Golde, and Alan Akbik. 2025. Sample-Efficient Language Modeling with Linear Attention and Lightweight Enhancements. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Valentin Hofmann, Leonie Weissweiler, David Mortensen, Hinrich Schütze, and Janet Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences of the United States of America*, 122:e2423232122.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyurek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of world knowledge (EWOK): A cognition-inspired framework for evaluating basic world knowledge in language models. *CoRR*, abs/2405.09605.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Despoina Kosmopoulou, Efthymios Georgiou, Vaggelis Dorovatas, Georgios Paraskevopoulos, and Alexandros Potamianos. 2025. Masked Diffusion Language Models with Frequency-Informed Training. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Nalin Kumar, Mateusz Lango, and Ondrej Dusek. 2025. Pretraining Language Models with LoRA and Artificial Languages. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Hyunji Lee, Wenhao Yu, Hongming Zhang, Kaixin Ma, Jiyeon Kim, Dong Yu, and Minjoon Seo. 2025. Understanding and Enhancing Mamba-Transformer Hybrids for Memory Recall and Language Modeling. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*.

Sharid Loáiciga, Eleni Fysikoudi, and Asad B. Sayeed. 2025. Exploring smaller batch sizes for a high-performing BabyLM model architecture. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yilan Lu. 2025. Navigating the Design Space of MoE LLM Inference Optimization. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Jonas Mayer Martins, Ali Hamza Bashir, Muhammad Rehan Khalid, and Lisa Beinborn. 2025. Once Upon a Time: Interactive Learning for Storytelling with Small Language Models. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Kate McCurdy, Katharina Christian, Amelie Seyfried, and Mikhail Sonkin. 2025. Two ways into the hall of mirrors: Language exposure and lossy memory drive cross-linguistic grammaticality illusions in language models. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Sushant Mehta, Raj Dandekar, Rajat Dandekar, and Sreedath Panat. 2025. Unifying Mixture of Experts and Multi-Head Latent Attention for Efficient Language Models. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Francesca Padovani, Bastian Bunzeck, Manar Ali, Omar Momen, Arianna Bisazza, Hendrik Buschmeier, and Sina Zarrieß. 2025. Dialogue Is Not Enough to Make a Communicative BabyLM (But Neither Is Developmentally Inspired Reinforcement Learning). In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Whitney Poh, Michael Tombolini, and Libby Barak. 2025. What did you say? Generating Child-Directed Speech Questions to Train LLMs. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *16th European Conference on Computer Vision*.

Rareș Păpușoi and Sergiu Nisioi. 2025. A Comparison of Elementary Baselines for BabyLM. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Matthew Theodore Roque and Dan John Velasco. 2025. Beyond Repetition: Text Simplification and Curriculum Learning for Data-Constrained Pretraining. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Yamamoto Rui and Keiji Miura. 2025. FORGETTER with forgetful hyperparameters and recurring sleeps can continue to learn beyond normal overtfitting limits. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Suchir Salhan, Hongyi gu, Donya Rooein, Diana Galvan-Sosa, Gabrielle Gaudeau, Andrew Caines, Zheng Yuan, and Paula Buttery. 2025a. Teacher Demonstrations in a BabyLM's Zone of Proximal Development for Contingent Multi-Turn Interaction. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Suchir Salhan, Richard Diehl Martinez, Zebulon Goriely, and Paula Buttery. 2025b. What is the Best Sequence Length for BabyLM? In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*.

Loris Schoenegger, Lukas Thoma, Terra Blevins, and Benjamin Roth. 2025. Influence-driven Curriculum

Learning for Pre-training on Limited Data. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sam Shleifer, Jason Weston, and Myle Ott. 2021. Normformer: Improved transformer pretraining with extra normalization. *CoRR*, abs/2110.09456.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Ece Takmaz, Lisa Bylinina, and Jakub Dotlacil. 2025. Model Merging to Maintain Language-Only Performance in Developmentally Plausible Multimodal Models. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Alexander Tampier, Lukas Thoma, Loris Schoenegger, and Benjamin Roth. 2025. RecombiText: Compositional Data Augmentation for Enhancing LLM Pre-Training Datasets in Low-Resource Scenarios. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Alvin Wei Ming Tan, Chunhua Yu, Bria Lorelle Long, Wanjing Anya Ma, Tonya Murray, Rebecca D. Silverman, Jason D Yeatman, and Michael Frank. 2024. DevBench: A multimodal developmental benchmark for language learning. In *The 38th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Joonas Tapaninaho. 2025. MoEP: Modular Expert Paths for Sample-Efficient Language Modeling. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jannek Ulm, Kevin Du, and Vésteinn Snæbjarnarson. 2025. Contrastive Decoding for Synthetic Data Generation in Low-Resource Language Modeling. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Dan John Velasco and Matthew Theodore Roque. 2025. Rethinking the Role of Text Complexity in Language Model Pretraining. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *The 33rd Conference on Neural Information Processing Systems*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. GIT: A generative image-to-text transformer for vision and language. *Transactions on Machine Learning Research*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*.

Jason Wei, Najoung Kim, Yi Tay, and Quoc Le. 2023. Inverse scaling can become U-shaped. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15580–15591, Singapore. Association for Computational Linguistics.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schütze, Kemal Oflazer, and David R. Mortensen. 2023. Counting the bugs in chatgpt's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6508–6524. Association for Computational Linguistics.

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger P. Levy. 2020. On the predictive

power of neural language models for human real-time comprehension behavior. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*, page 1707–1713.

Ethan Gotlieb Wilcox, Michael Y Hu, Aaron Mueller, Alex Warstadt, Leshem Choshen, Chengxu Zhuang, Adina Williams, Ryan Cotterell, and Tal Linzen. 2025. Bigger is not always better: The importance of human-scale language modeling for psycholinguistics. *Journal of Memory and Language*, 144:104650.

Ethan Gotlieb Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy. 2023. Testing the Predictions of Surprisal Theory in 11 Languages. *Transactions of the Association for Computational Linguistics*, 11:1451–1470.

L'uboš Kriš and Marek Suppa. 2025. SlovakBabyLM: Replication of the BabyLM and Sample-efficient Pretraining for a Low-Resource Language. In *Proceedings of the 3rd BabyLM Challenge at the 2025 Conference on Empirical Methods in Natural Language Processing*.

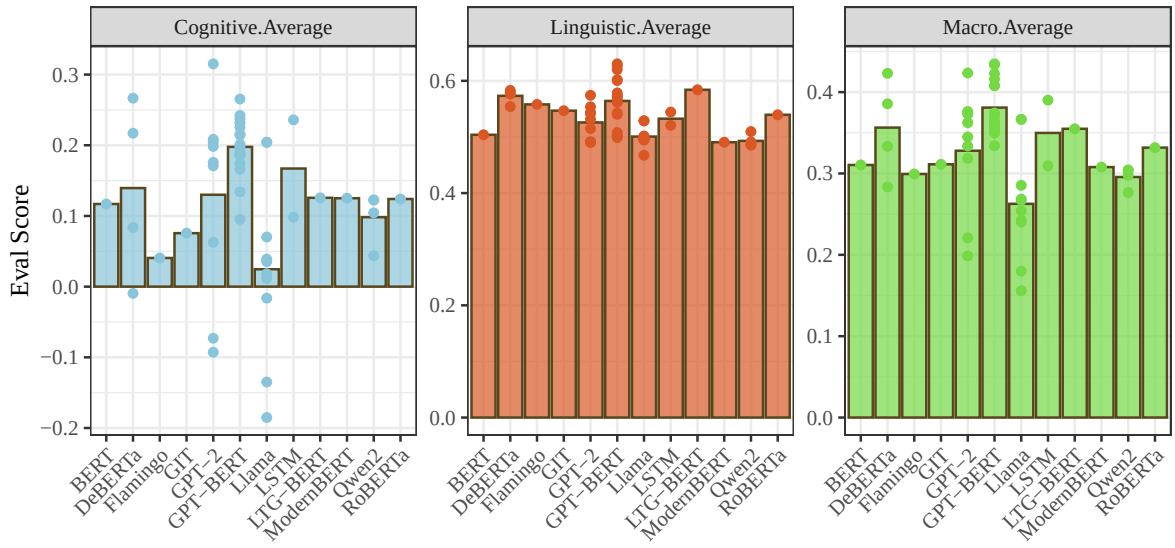


Figure 7: **Scores by backbone architecture** As with last year, we find that the GPT-BERT model consistently leads to stronger performance. Also consistent with previous years, we find that DeBERTa and LTG-BERT lead to strong performance as well.

A Additional Results

B HuggingFace Repository Names for Baseline Models

Model	HuggingFace Repository
STRICT	
GPT-BERT-causal	BabyLM-community/babylm-baseline-100m-gpt-bert-causal-focus
GPT-BERT-mixed	BabyLM-community/babylm-baseline-100m-gpt-bert-mixed
GPT-BERT-masked	BabyLM-community/babylm-baseline-100m-gpt-bert-masked-focus
GPT2	BabyLM-community/babylm-baseline-100m-gpt2
STRICT-SMALL	
GPT-BERT-causal	BabyLM-community/babylm-baseline-10m-gpt-bert-causal-focus
GPT-BERT-mixed	BabyLM-community/babylm-baseline-10m-gpt-bert-mixed
GPT-BERT-masked	BabyLM-community/babylm-baseline-10m-gpt-bert-masked-focus
GPT2	BabyLM-community/babylm-baseline-10m-gpt2
MULTIMODAL	
Flamingo	BabyLM-community/babylm-multimodal-baseline-flamingo
Git	BabyLM-community/babylm-multimodal-baseline-git
INTERACTION	
SimPO	BabyLM-community/babylm-interaction-baseline-simpo

Table 4: HuggingFace Repositories for the baseline models separated by tracks.

C Interaction External Model Correction Prompt

The prompt used for the external model to correct student model generations is shown in Figure 9.

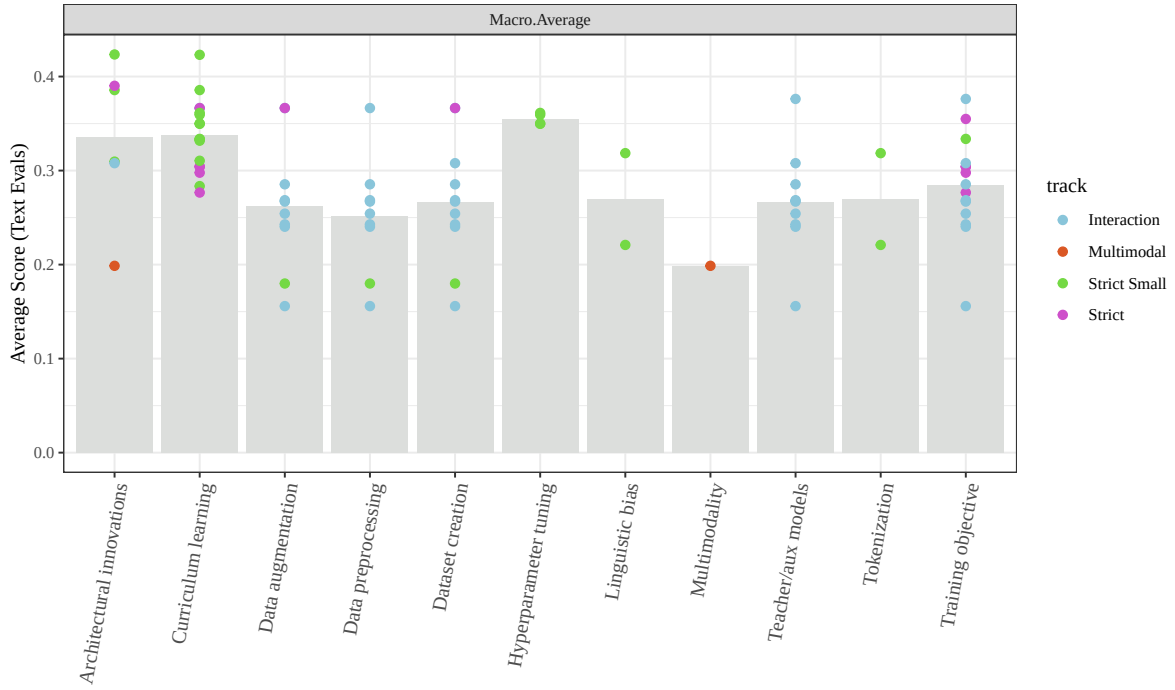


Figure 8: **Scores by approach taken** As with previous years, we find higher scores for models that employ architectural innovations. We also find higher scores for models that employ hyperparameter tuning.

Correction Prompt:
[User] You will be given a partial text (labeled "Partial Text") and a completion of said text produced by a student of English (labeled "Student Completion"). Your goal is to produce a corrected version of the student's completion. This corrected version should be grammatically correct, coherent and relevant to the initial partial text. If the student's response is incomprehensible, output your own independent completion. You should only provide your own completion without any added commentary or feedback.

Partial Text: <student input>
Student Completion: <student completion>

Now produce your own completion of the Partial Text. Do not include any external commentary.

[Assistant] Partial Text: <student input>
Corrected Completion:

Figure 9: The prompt given to the teacher model to sample corrected versions of the student's completions.

D Detailed Findings across the BabyLM Workshop and Challenge Submission

We synthesized results across the Workshop and Challenge tracks, examining each paper in terms of its data operations, training and optimization strategies, architectural choices, evaluation dimensions, release artifacts, developmental plausibility, multilingual scope, and use of MoE/sparsity mechanisms, along with additional factors including interaction and feedback methods, tokenizer family, objective variants, data provenance, selection policies, and competence-related effects. At BabyLM scales (≈10M-100M tokens), small design choices, such as tokenization, sequence length, or optimizer cadence, often yielded gains comparable to or greater than architectural modifications (Salhan et al., 2025b). Below we summarize the dominant empirical patterns that emerged across submissions.

D.1 Data: Selection Over Ordering, and the Shape of "Helpful" Synthetic Data

Selection Outperforms Human-Curated Ordering. Across multiple studies, model-driven selection criteria proved substantially more effective than human-designed curricula or naïve ordering strategies. Influence-based curricula (Schoenegger et al., 2025) improved performance by prioritizing examples that have the greatest effect on model predictions, while active surprisal-based selection (Fysikoudi et al.,

2025) dynamically focuses training on inputs the model finds most uncertain. Notably, gains arise not from a particular ordering direction (e.g., easy-to-hard), but from grouping data by similar influence or uncertainty levels, which stabilizes learning dynamics and improves generalization under strict data constraints.

Simplification of corpus text helps when balanced with diversity and aligned with model capacity. In several submissions, LLM-assisted simplification of existing corpus text improved sample efficiency and accelerated convergence under constrained token budgets. However, this was only true when simplification was applied as *augmentation* rather than replacement (Velasco and Roque, 2025; Roque and Velasco, 2025). Smaller models (under 200M parameters) benefited from simple-to-complex curricula, consistent with classical "starting small" effects; in contrast, relatively larger BabyLM-scale models (300M-1B) achieve higher downstream accuracy when simplified and original text were *interleaved*, indicating that simplification operated as a form of regularization.

Conversely, simplification improved linguistic knowledge transfer and zero-shot generalization only when the diversity and semantic coverage of the original corpus were preserved. Narrow simplification strategies that target a single linguistic feature like inserting only pedagogical questions, led to overfitting towards stylistic cues and reduced robustness on evaluation tasks (Poh et al., 2025). This shows that simplification is not inherently beneficial: its value lies in enhancing *coverage density* rather than constraining style.

Synthetic data is most effective when it complements rather than replaces natural text. Across both the tracks, two forms of synthetic augmentation consistently improved performance under fixed token budgets. Contrastive synthetic data, which is generated using paired Good/ Bad completions improved reasoning and robustness more reliably than vanilla synthetic sampling by providing explicit discriminative signals (Ulm et al., 2025). The effectiveness of this approach depended critically on maintaining diversity and balancing synthetic and natural data. Compositional, corpus-internal augmentation strategies, such as recombining semantically compatible sentence fragments, improved entity tracking, morphology, and several NLU metrics when synthetic data made up approximately half of the pretraining corpus (Tampier et al., 2025). Performance declined when synthetic data dominated the corpus, underscoring the need to ground augmentation in authentic linguistic distributions.

Submissions to both tracks converged on the same principle: hybrid data regimes, where synthetic and natural text were interleaved or mixed in controlled ratios, consistently outperformed purely synthetic or purely natural corpora at BabyLM scale.

D.2 Objectives and Training: Small Knobs, Big Effects

A consistent theme across submissions was that modifying the learning objective itself often yielded gains comparable to scaling data or model size. Several papers demonstrated that the choice and scheduling of the pretraining loss function directly shaped sample efficiency and downstream generalization. For example, *Mask and You Shall Receive* (Edman and Fraser, 2025) introduced an adaptive masked language modeling objective in which harder-to-predict tokens were masked more frequently, leading to improved performance on morphology-sensitive evaluations such as the WUG test. Similarly, *Babies Learn to Look Ahead* (Aynetdinov and Akbik, 2025) showed that incorporating multi-token prediction improved entity tracking and discourse modeling in models as small as 130M parameters trained on 10M tokens, with curriculum-based scheduling outperforming static variants. Alternative objectives were also explored, including diffusion-based language modeling (Kosmopoulou et al., 2025), where frequency-aware noise schedules yielded performance competitive with GPT-BERT hybrids, and parameter-efficient strategies such as pretraining on artificial formal languages followed by LoRA adaptation (Kumar et al., 2025), which outperformed full-parameter training on morpho-syntactic benchmarks.

Another set of findings showed that training cadence and procedural hyperparameters can rival architectural changes in their effect on model quality. *Exploring Smaller Batch Sizes* (Loáiciga et al., 2025) reported that reducing effective batch size while using gradient accumulation improved generalization on BLiMP and MSGS benchmarks, suggesting that increased optimization noise benefits small-data learning.

Complementary work (Rui and Miura, 2025) demonstrated that periodically resetting optimizer states allowed models to continue improving beyond conventional convergence points, yielding lower validation loss on both Baby10M and Baby100M settings. Additionally, *What's the Best Sequence Length?* (Salhan et al., 2025b) found that the optimal sequence length was highly dependent on both architecture and task: longer contexts benefited analogy and entity tracking tasks in state-space models, whereas shorter sequences were sufficient for syntactic generalization in transformer-based architectures.

D.3 Architectures: Efficiency, Linear-Time Models, and Sparse Routing Mechanisms

One key finding across submissions is that architectural modifications which reduce attention complexity or introduce structured sparsity often yield measurable gains under our strict data and compute constraints, particularly when paired with appropriate optimization. Linear and state space model (SSM)-based token mixers acted as alternatives to full self-attention. Haller et al. (2025) replaced self-attention with an mLSTM token mixer, combining them with lightweight modifications such as sliding-window attention and short convolutions, improved zero-shot performance and training stability, especially when using the Muon optimizer instead of AdamW. Lee et al. (2025) further demonstrated that hybrid architectures combining state space models with attention yield complementary strengths: sequential hybrid architectures performed better on short-context tasks, while parallel architectures with cross-attention achieved better long-context recall.

Sparse and routed architectures were also investigated from both deployment and learning perspectives. On the systems side, *Navigating the Design Space of MoE Inference Optimization* (Lu, 2025) evaluated expert offloading, quantization, and distillation strategies for serving mixture-of-experts (MoE) models under memory and latency constraints, finding that dynamic expert offloading can maintain model quality while reducing hardware requirements. On the learning side, other work (Tapaninaho, 2025) introduced token-routed sparse paths across modular transformer blocks and reported faster early learning and improved strict-small benchmark scores relative to a dense GPT-2 baseline, though with later-phase stability trade-offs. Mehta et al. (2025) presented a combined MoE and latent attention architecture that reduced KV-cache memory while maintaining competitive perplexity, suggesting that MoE-style routing and compression mechanisms can be jointly leveraged to improve efficiency.

D.4 Tokenization and Morphology

A consistent pattern across both Workshop and Challenge submissions is that tokenization choices exert disproportionately large effects in our constrained setups (*Strict*, *Strict-Small*), often rivaling objective or architectural modifications. Models using morphology-aware tokenizers demonstrated substantial gains in entity tracking and world knowledge tasks. One submission that compared BPE with rule-based and unsupervised morphological tokenization reported improvements of approximately 20% on EWoK and 40% on entity tracking when morpheme segmentation was applied, indicating that linguistically grounded token boundaries directly support better generalization in small models (Bölücü and Can, 2025). Curriculum-based introduction of morphology yielded mixed results: it added modest improvements for GPT-BERT architectures but degraded BLiMP performance in GPT-2 variants.

Multiple papers evaluated the redistribution of linguistic competence induced by tokenizer choice. Systems trained with BPE typically achieved the strongest syntactic acceptability judgments, while morphology or syllable-aware tokenizers improved semantic generalization and discourse tracking (Fusco et al., 2025; Păpușoi and Nisioi, 2025). These findings show that tokenization implicitly moves models toward different linguistic capacities, even when architecture and training data are held constant.

Evidence from multilingual model training with morphologically rich languages further supported the role of tokenization. In Hebrew, a compact RoBERTa-style model trained with morphology-aware representations achieved competitive grammatical judgments despite a reduced data budget (Gelboim and Sulem, 2025). In Slovak, a replication study found that token inflation caused by applying English-trained BPE tokenizers, increased the number of tokens per sentence and effectively reduced the usable data budget. In this setting, tokenization appeared as the single highest-leverage intervention, surpassing curriculum and architecture in impact (L'uboš Kriš and Suppa, 2025).

D.5 Interaction, Feedback, and Alignment: Learning Beyond Pretraining Tokens

A small proportion of submissions that targeted interaction or feedback showed that alignment signals can act as efficient substitutes for large-scale pretraining. One line of work focused on dialogue alignment using minimal preference pairs. Padovani et al. (2025) show that fine-tuning with Direct Preference Optimization (DPO) on child-caregiver dialogue minimal pairs improved pragmatic choice behavior, leading to higher accuracy on communicative benchmarks, even though zero-shot language modeling metrics such as BLiMP and EWoK remained unchanged. In contrast, Proximal Policy Optimization (PPO), had mixed effects and occasionally destabilized model behavior. Another submission framed narrative generation as an interactive learning problem. In the storytelling setup, a teacher model assigned feedback on readability, coherence, and creativity to student-generated stories (Martins et al., 2025). With fewer than one million interactive tokens, the student model achieved gains comparable to a model trained on 100M tokens, particularly in narrative cohesion and entity tracking, while retaining performance on formal linguistic benchmarks.

A related submission introduced teacher demonstrations as aligned continuations in multi-turn dialogue. Salhan et al. (2025a) showed that models trained on "edited" responses provided by a teacher language model produced more contextually contingent and cohesive dialogue turns than baseline autoregressive models. This work further demonstrated that post-training on preference pairs improved multi-turn interaction quality without requiring large additional corpora. Together, these studies show that structured interaction primarily implemented through preference alignment (reinforcement-style scoring, or teacher demonstrations) helps induce qualitative gains in communicative and functional language use at a fraction of the token cost of additional pretraining.

D.6 Evaluation Beyond Grammar: Pragmatics, Learner Profiles, Developmental Trajectories, and Multimodal Trade-offs

Some submissions expanded the BabyLM evaluation landscape beyond traditional grammatical benchmarks, introducing new metrics for assessing pragmatic competence, second-language developmental profiles, learning trajectories, and multimodal efficiency.

Several works evaluated models on pragmatic reasoning. In one of the outstanding papers, BabyLM-scale models were assessed on a benchmark grounded in Gricean maxims (Askari et al., 2025). The authors found that while models trained on 100M tokens outperformed those trained on 10M, all models lagged behind child-level performance, with the largest deficits observed in the maxim of Quantity, indicating continued difficulty in evaluating informativeness. This pattern held even when other maxims, such as Quality and Relation, and showed moderate improvement with scale.

Another direction evaluated models through the lens of second-language acquisition. The BLiSS benchmark (Gao et al., 2025) introduced a large-scale evaluation of learner-like grammatical competence, using minimal pairs derived from annotated L2 corpora and organized by CEFR proficiency level and learner L1. Models were assessed on their ability to distinguish learner errors from corrections, revealing systematic differences across training regimes. Tokenization choice and transfer learning strategies significantly influenced alignment with bilingual learner profiles.

Developmental comparisons were explicitly explored in work on determiner acquisition trajectories. One submission (Fiandra et al., 2025) compared intermediate training checkpoints of BabyLM models with speech samples from children, showing that children consistently produced indefinite determiners first, while models acquired definite determiners earlier. This divergence suggests that models optimize for frequency and predictability rather than cognitive developmental salience. McCurdy et al. (2025) demonstrated that both language exposure statistics and memory constraints contribute to model behavior, but neither factor alone accounted for human-like processing across languages.

Finally, multimodal submissions examined the interaction between vision-language grounding and linguistic competence. One study showed that multimodal pretraining reduced performance on text-only grammatical benchmarks, but that merging the parameters of a multimodal model with those of a text-only model through weighted interpolation partially restored language-focused performance while maintaining multimodal capabilities (Takmaz et al., 2025). Another submission introduced a low-bit multimodal

fusion model with episodic memory (AMAN et al., 2025), demonstrating that aggressive quantization and memory augmentation allowed on-device deployment while preserving basic multimodal reasoning, albeit with reduced fine-grained linguistic fidelity.

Collectively, these evaluations extend the BabyLM paradigm beyond formal grammatical competence, revealing distinct dimensions of pragmatic inference, bilingual developmental alignment, cognitive trajectory modeling, and multimodal trade-offs.

D.7 Challenge versus Workshop Contributions

The Challenge track primarily focused on mechanisms for increasing sample efficiency under fixed token constraints. Submissions explored adaptive objectives such as difficulty-aware masked language modeling and multi-token prediction, as well as diffusion-based language modeling, demonstrating that modifying the learning signal can recover performance otherwise dependent on larger datasets. Several entries adopted parameter-efficient pretraining strategies, including the use of artificial structural priors followed by LoRA adaptation, and others demonstrated that small-scale interactive feedback could enhance communicative behavior with fewer than one million additional tokens. Corpus-internal augmentation methods, such as compositional recombination, were also introduced as an alternative to external synthetic generation, enabling performance gains while adhering to the Challenge's data budget constraints.

By contrast, the Workshop track broadened the evaluation and systems landscape. Submissions introduced new axes of measurement beyond grammatical competence, including pragmatic informativeness, bilingual learner profiles, and developmental trajectories derived from longitudinal human acquisition data. Other work focused on architectural and deployment efficiency, exploring sparse mixture-of-experts models, latent attention mechanisms, multimodal alignment, and model merging techniques designed to restore linguistic competence after multimodal pretraining.

Contributions to both tracks are in spirit of the BabyLM goals: rethinking data use rather than increasing data volume.

Dialogue Is Not Enough to Make a Communicative BabyLM (But Neither Is Developmentally Inspired Reinforcement Learning)

Francesca Padovani[1]* **Bastian Bunzeck**[2]* **Manar Ali**[2] **Omar Momen**[2]
Arianna Bisazza[1] **Hendrik Buschmeier**[2] **Sina Zarrieß**[2]
[1]Center for Language and Cognition (CLCG), University of Groningen
[2]CRC 1646 – Linguistic Creativity in Communication, Bielefeld University
f.padovani@rug.nl bastian.bunzeck@uni-bielefeld.de

Abstract

We investigate whether pre-training exclusively on dialogue data results in formally and functionally apt small language models. Based on this pre-trained `llamalogue` model, we employ a variety of fine-tuning strategies to enforce "more communicative" text generations by our models. Although our models underperform on most standard BabyLM benchmarks, they excel at dialogue continuation prediction in a minimal pair setting. While PPO fine-tuning has mixed to adversarial effects on our models, DPO fine-tuning further improves their performance on our custom dialogue benchmark.

1 Introduction

Large language models are capable of generating language with almost human-like fluency. To do so, however, they need unfathomable amounts of textual input as training data. In comparison, humans are highly sample-efficient learners and develop a full-fledged linguistic system from input that is orders of magnitude smaller. In the past, this sample efficiency has mostly been attributed to genetically pre-endowed priors (Chomsky, 1986; Berwick et al., 2011). More recently, the quantitative, usage-based turn in linguistics has focused on the importance of language use, interaction and grounding in the real world and more domain-general cognitive mechanisms for language learning (Tomasello, 2003, 2005; Behrens, 2021). Crucially, language is primarily a tool for communication (Fedorenko et al., 2024; Levinson, 2025), and therefore all acquisition processes must be conceptualized accordingly.

Lately, the BabyLM paradigm has emerged as a novel way of testing claims of learnability with little data, small language models and linguistically inspired evaluation tasks (Warstadt et al., 2023; Hu et al., 2024; Charpentier et al., 2025). Although highly optimized models are indeed able to capture linguistic structure very accurately (e.g., Charpentier and Samuel, 2023; Tastet and Timiryasov, 2024), they are still trained on a wider variety of input registers than the main input modality of children, namely child-directed speech in dialogue. Observed in isolation, child-directed speech does differ tremendously from other input modalities, featuring many fragments, more questions and less canonical SV(X) sentences (Cameron-Faulkner et al., 2003; Bunzeck and Diessel, 2025). Despite Huebner et al. (2021) finding it to be conducive pretraining data for simplified benchmarks, more recent work has shown that its effects can be described as *mixed* at best (Padovani et al., 2025; Bunzeck et al., 2025).

One possible explanation for this discrepancy is that autoregressive language models, trained on a next-token prediction task, do not model the communicative aspects that are seen as crucial for language acquisition and underlie the fragmented nature of child-caregiver dialogue. Common data pre-processing protocols for BabyLMs split child-caregiver dialogues into isolated sentences, which effectively removes communicative context that is available and central for human learners. Therefore, we conceptualize the task of training a BabyLM differently: We train a small, autoregressive model[1] on dialogue triplets extracted from CHILDES (MacWhinney, 2000). As such, our model is not a model of the learner *per se*, but of the interaction and communication underlying the language learning process. Additionally, we apply different reinforcement learning paradigms to our model to make the 'child' component of the dialogue system more fluent and contextually appropriate when interacting with a 'caregiver' dialogue partner. In sum, we test the following ideas through this process: (i) How does a BabyLM trained only on child–caregiver dialogue perform?

*These authors contributed equally.

[1]Given its training on dialogue data only, throughout this paper we refer to the base model as `llamalogue`. All models and datasets can be found in this Huggingface collection.

Proceedings of the First BabyLM Workshop, pages 433–447
November 8, 2025

And (ii) Are there ways of teaching BabyLMs to be more communicative speakers via interaction and communication?

We find that (i) our base model pre-trained exclusively on child–caregiver dialogues maintains above-chance accuracy on formal linguistic competence, while achieving higher accuracy in predicting realistic communicative turns than a baseline autoregressive model. Moreover, (ii) directly aligning preferred child responses to caregiver utterances through DPO proves more effective than interactively fine-tuning the policy via PPO with a reward function, especially when evaluated on dialogue minimal pairs. However, none of these fine-tuning techniques improves performance on more formal benchmarks.

2 Related work

Learning exclusively from CDS While the standard English BabyLM corpus consists of approximately 30% child-directed speech, ample work exists on pretraining LMs from scratch on 100% child-directed speech (CDS). In a seminal paper, Huebner et al. (2021) showed that a small 5M-parameter BabyBERTa model, trained on 5M lexical tokens of child-directed speech, shows the same accuracy on Zorro (vocabulary-limited minimal pair tasks; Huebner et al., 2021) as the RoBERTa-base model with 125M parameters and trained on 30B words. Similar results are presented by Feng et al. (2024), who show that autoregressive models trained on CDS alone perform only slightly worse on Zorro than comparable architectures trained on Wikipedia data, synthetic data, or the BabyLM corpus. However, their CDS models underperform other models tremendously on semantic similarity benchmarks. Negative results are also reported by Yedetore et al. (2023), who show that autoregressive models trained on CHILDES data fail to reliably acquire hierarchical generalizations in question formation from declaratives, and rather prefer incorrect linear generalizations.

Expanding the CDS-only training paradigm to more languages than English, Salhan et al. (2024) find that developmentally-inspired curriculum learning strategies during pretraining improve scores on syntactic minimal pairs for models trained on English, French, German, Chinese, or Japanese CDS, outperforming models trained on Wikipedia data by over 10%. Conversely, Padovani et al. (2025) report less positive results. For many syntactic minimal pair benchmarks, their CDS models underperform in comparison to Wikipedia-trained models across different languages (English, German, French). Finally, Bunzeck et al. (2025) approximate German CDS on the level of utterance-level construction distributions. They also find that models trained on it are generally inferior to models trained on comparable Project Gutenberg data when evaluated on syntactic benchmarks, although the CDS models show moderate improvements on some word-level benchmarks.

In sum, it can therefore be said that pre-training on CDS is only conducive to language model performance for highly specific benchmarks like Zorro (although results are inconsistent across studies) or in more specific training regimens like curriculum learning.

Cognitively/developmentally plausible RL Despite reinforcement learning, especially in the form of RLHF (Ouyang et al., 2022), being an integral part of modern language modeling practices, it has only very recently begun to get adopted in cognitively inspired modeling. Zhao et al. (2023) improve their small models trained on BabyLM data by constructing a RLHF dataset from human-annotated story continuations generated with regular GPT-2 and then reinforcing these storytelling capabilities of their models. While it does not improve performance on zeroshot benchmarks, it makes their models better base models for fine-tuning tasks.

In a more developmentally inspired fashion, Ma et al. (2025) generate text continuations from a student GPT-2 model and compare these to an already further trained teacher model. A reward signal is then generated from the model's estimated 'age' (*viz.* training steps), based on its continuations and the teacher continuation. This interactive learning is then interleaved with regular causal language modeling. Their interactive model outperforms regular autoregressive models on word acquisition, quantified as average surprisal for a set of test sentences.

Stöpler et al. (2025) introduce a training regime inspired by emergent communication research that again includes two language models: a speaker/child language model, and a listener/caregiver language model. In their setup, the speaker model has to summarize a passage, and the listener model has to answer a question solely based on the summary provided by the speaker. If the listener model (whose weights are frozen) answers correctly, a reward signal is used to update the speaker model. Although

their reinforcement strategy changes speaker behavior, it does not improve performance on linguistic benchmarks.

Finally, Nikolaus and Fourtassi (2025) base their reward signal on an annotated dataset of CHILDES data with clarification requests in parental utterances, which often trigger children to use more "grammatical" language. For each utterance produced by their child language model, they predict if it would possibly beget such a clarification request, and reward productions that do not. They show that this process improves their models on the reinforcement goal of producing less "ungrammatical" utterances, but has mixed to no effects on grammar benchmarks like BLiMP (Warstadt et al., 2020) and Zorro.

3 Methodology

3.1 Pretraining

Data Our models are trained on dialogue data from the English CHILDES section. In a first preprocessing step, we clean the transcripts from CHILDES quite heavily by removing all extra- and paralinguistic information. Furthermore, we replace all unintelligible or otherwise incomplete utterances, for which annotations as to the intended word are available, with these intended words. Finally, we split all utterances that contain explicitly annotated pauses, as there is no clear distinction between such pauses and utterance boundaries marked by regular line breaks.

From these cleaned dialogues, we extract all utterance triplets (three consecutive turns) where at least two different speakers are involved. Furthermore, we enforce the triplets to contain at least five lexical words. This excludes triplets that only contain repetitions of single words or are otherwise light on lexical content. We leave the speaker tags in the data. A typical line from our data might therefore look as follows:

```
*CHI: all gone .
*MOT: where's the kitty ?
*CHI: all gone .
```

By using dialogue data only, we assume that the autoregressive pretraining process pushes our BabyLM to model contingent structure (responses depend on previous turns), learn turn-level coherence, and acquire some knowledge about implicit expectations in communication, e.g., that questions beget a response.

Base model We train a small 135M-parameter Llama model (Touvron et al., 2023a) on 10M lexical tokens from the aforementioned set of dialogue triplets. Our model features 16 layers, 16 attention heads and a hidden/intermediate size of 1024. We train the model for 10 epochs. As we found approximately 60k different lexical types in our data, we opt for a small vocabulary size to not store too many of these types holistically. We fit a BPE tokenizer on the training data to include 8k tokens. Crucially, we fit the tokenizer on the actual transcriptions only, not on the speaker tags. The speaker tags are added as additional tokens afterwards. In sum, with the inclusion of all speaker tags, this results in a vocabulary size of 8465.

3.2 DPO fine-tuning

As a first attempt to further align `llamalogue` with child-like, communicatively appropriate behavior, we employ Direct Preference Optimization (DPO; Rafailov et al., 2023), which is a preference-based training method that directly optimizes the model to prefer certain continuations over others. In our case, this procedure is supposed to guide our model to favor contextually appropriate utterances over random ones.

Naturalistic data As fine-tuning data, we construct a dataset of minimal dialogue pairs derived from another set of triplets not seen during pretraining and not used for validation. From these, we extract naturally occurring caregiver–child exchanges and derive contrastive, incorrect variants by replacing the child utterance with a randomly sampled one. To systematically control for confounds, we focus on minimal pairs that are matched in length (by number of words or subword tokens) and filter out pairs where the child utterance repeats words from the caregiver utterance, resulting in approximately 26 000 pairs. For DPO training, we select the word-matched minimal pairs, subsampling 18 000 examples for training, with the remaining 8000 examples held out for evaluation. Overall, the fine-tuning phase was conducted on a total of 245 480 tokens.

Synthetic data In addition to the real data, we generate a synthetic DPO dataset to probe the benefits of model-guided preference generation. Here, the caregiver's utterance is used as a prompt to Llama-3.2-3B (Touvron et al., 2023b), which generates a plausible child response. Incorrect alternatives are again randomly sampled from the original dataset.

You are a young child having a conversation with your mother.
When your mother says something, you should answer as a typical and natural-sounding child. Do NOT repeat her words. Instead, give a new, relevant answer that shows understanding.
Keep it short and child-like.

*MOT: I think they just throw it on the side .
*CHI:

Table 1: Zero-shot prompt to Llama-3.2-3B.

Here, we do not control for matched length, as the exact number of generated words is not easy to control. The child continuation is generated through an instructive prompt (cf. Table 1) designed to facilitate short and natural completions. In total, the synthetic training data is composed of 245 480 tokens.

The two fine-tuning datasets are available in our Huggingface collection. Representative examples from both datasets are provided in Appendix B.

We perform one 10-epoch DPO fine-tuning run with `llamalogue` on each dataset with `trl` (von Werra et al., 2020). A learning rate of 5×10^{-6} is used, with a per-device batch size of 4, and 4-step gradient accumulation, resulting in an overall batch size of 16. Figure 1 shows the two loss and reward trends for the appropriate and random sentences of the fine-tuning datasets.

3.3 PPO fine-tuning

To steer the communicative behaviour of `llamalogue` more indirectly, we also fine-tune it using Proximal Policy Optimization (PPO; Schulman et al., 2017). To implement the notion of 'effective communication' for PPO, we needed to substantially simplify it. Developmental research has extensively characterized learning as involving a dynamic exploration–exploitation trade-off (Kim and Carlson, 2024; Gopnik, 2020; Nussenbaum and Hartley, 2019), in which children alternate between experimenting with novel behaviors (i.e., linguistic forms) and leveraging familiar patterns. However, operationalizing this *sweet spot* between exploration and exploitation as a computational reward function is inherently difficult. To formalize what constitutes a "successful" communicative turn, we explored a range of reward functions reflecting different aspects of communication: a BLEU-based reward, a semantic similarity reward, a quality score derived from an LLM, and an uncertainty-based reward measuring LLM confidence in processing child responses.

Our PPO pipeline requires real caregiver prompts as input for both `llamalogue` and a teacher LLM emulating a "good, communicative baby", therefore we extract caregiver utterances (minimum four tokens length) from unused segments of the preprocessed CHILDES dialogue triplets. We then prompt a teacher LLM, as before a Llama-3.2-3B (Touvron et al., 2023a), with these utterances, asking it to generate candidate responses simulating a short child-like answer that shows understanding of the caregiver utterance[2]. The prompt is the same as the one used for generating the DPO datasets (Table 1). The reward functions are then computed by comparing these teacher-generated responses to the output produced by `llamalogue` in response to the same utterance. Calculating the reward as an average over 10 generated responses proved to be noisy due to their variability, so we ultimately based the reward on the comparison between the model's output and the one single response generated by the teacher LLM.

1-gram BLEU Reward The BLEU-based metric (Papineni et al., 2002) captures surface-level lexical similarity. Specifically, we compute a smoothed unigram BLEU score (BLEU-1) between `llamalogue`'s response and the teacher LLM's reference answer with `nltk`. We apply smoothing to avoid zero scores. The resulting reward values range from 0 to 1.

Semantic Similarity Reward As a complementary approach to lexical overlap, we also implement a semantic similarity reward to promote contextually appropriate, meaningful responses. Specifically, we use the `all-MiniLM-L6-v2` model from `SentenceTransformers` (Reimers and Gurevych, 2019) to compute the cosine similarity between the BabyLM's response and the reference utterance generated by the teacher LLM. This similarity score, ranging from 0 to 1, encourages outputs that align semantically with high-quality examples.

LLM-generated Reward To further explore reward signals grounded in communicative quality, we prompt an LLM to directly assess `llamalogue`'s responses. Given a caregiver utterance and the generated child continuation of `llamalogue`, the LLM is instructed to assign a numerical quality score (from 0 to 5) based on contextual appropriateness and

[2]Examples can be found in Appendix C.

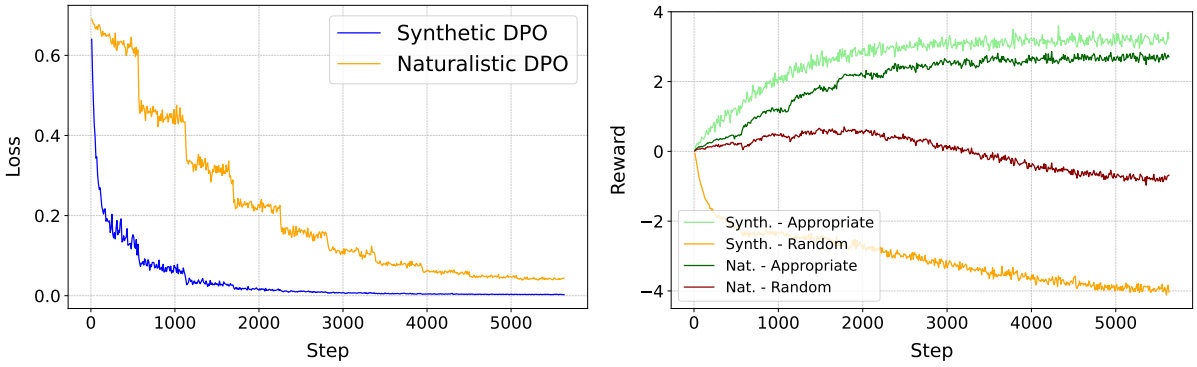


Figure 1: Training loss (left) and reward trends (appropriate vs. random) during training (right) for both DPO models.

<|system|> You are presented with a dialogue between a mother (MOT) and a child (CHI). Please rate how contextually appropriate and fluent the child's response is, on a scale from 0 (completely unfitting) to 5 (perfectly fine answer). If CHI answer is too short rate it low. <|end|> <|user|> MOT: It's like in the grocery store when go shopping . CHI: Mom, please let me choose the food for myself. <|end|> <|assistant|>

Table 2: Zero-shot prompt to OLMo.

fluency (see Table 2). After experimenting with various models, including Llama-3.2-3B and Nemotron-Research-Reasoning-Qwen-1.5B (Liu et al., 2025), we selected OLMo-2-1124-7B-Instruct (Olmo et al., 2025). This choice was motivated by the fact that OLMo consistently adhered to the requested output schema and avoided formatting anomalies that hindered automated reward extraction. The scalar score returned by OLMo is then used as the reward signal during each PPO step.

Teacher Confidence-based Reward To incorporate a measure of uncertainty into the reward signal, we implement a confidence-based metric: For each caregiver utterance x, we precompute the log-probabilities $\{\ell_i\}_{i=1}^{10}$ assigned by the frozen Llama-3.2-3B to the set of 10 reference child responses generated by that same model $\{y_i\}_{i=1}^{10}$, as explained in Section 3.3. During fine-tuning, the BabyLM's generated response $\tilde{y}$ is scored using the same teacher to obtain

$$\ell_{\text{baby}} \;=\; \log P_{\text{teacher}}(\tilde{y} \mid x)$$

Then we compute the normalized rank

$$\text{rank}(x, \tilde{y}) \;=\; \frac{1}{10} \sum_{i=1}^{10} \mathbf{1}\{\ell_i \le \ell_{\text{baby}}\} \in [0, 1],$$

and linearly map it to a PPO reward

$$r(x, \tilde{y}) \;=\; 2\,\text{rank}(x, \tilde{y}) - 1 \;\in [-1, 1].$$

This signal favors BabyLM outputs that the teacher assigned high likelihood and potentially bias the model towards more grammatical and distributionally expected utterances.

Training configuration In our experimental trials we rely on the default PPO training parameters provided by the `trl` library for all fine-tuned models, with the exception of the one trained using the Teacher Confidence-based reward. This reward caused higher variance in the reward values, making the KL control more sensitive. Therefore we set the KL penalty mode to `abs`, a lower learning rate of 5×10^{-6} and a small initial KL coefficient of 0.02 to weaken the penalty for policy updates in the early stage of training.

Moreover, the fine-tuning processes based on the first three PPO strategies employed a larger portion of the training data, as caregiver utterances inputs, from our original pre-processed set (220 000) compared to the model fine-tuned using the Teacher Confidence-based reward (150 000). In the latter case, we observed that a lower number of training steps was sufficient to achieve a consistent, significant improvement in the reward. We fine-tune for 3 epochs, with each epoch featuring 13 750 steps for the first three PPO strategies and 8645 steps for the Teacher Confidence-based reward. In terms of token usage, for the first three PPO strategies we estimate a total of 3 009 104 tokens, obtained by summing the

	Task	llamalogue	DPO		PPO				Baseline
			Natural.	Synth.	Bleu	SemSim	LLM Score	Conf.	
Zero-shot (Baby LM)	BLiMP	56.05	55.64	55.51	55.14	**56.36**	55.31	55.10	72.16
	BLiMP suppl.	51.06	49.97	**51.67**	51.33	51.48	50.58	49.45	61.22
	COMPS	51.62	51.51	**51.63**	50.66	51.58	51.25	51.59	—
	Entity tracking	30.66	32.66	31.29	16.20	34.64	**36.03**	34.05	28.06
	EWoK	50.19	50.12	**50.82**	49.65	49.62	50.12	50.81	51.92
	Read. (eye track.)	**3.88**	3.57	1.16	3.43	2.85	3.73	3.35	9.08
	Read. (self-paced)	1.43	1.35	0.44	**1.99**	1.04	1.30	1.14	3.5
	Wug adj.	0.45	0.52	0.16	0.13	0.01	**0.55**	0.41	38.5
	Wug past	-0.03	-0.01	-0.05	-0.15	-0.18	-0.01	-0.19	—
	AoA	-79.6	0	0	-80.1	0	-76.6	-78.7	—
FT	(Super)GLUE	51.82	51.72	51.77	51.12	52.10	51.69	**51.92**	67.91
Zero-shot (Add'l)	Lexical decision	40.3	40.5	**41.3**	40.7	39.7	40.2	40.8	57.2
	Zorro	**65.5**	64.8	62.7	62.5	64.7	65.2	63.7	77.7
	Dia. MP (Words)	64.3	**68.4**	64.9	62	61.1	60.6	63.7	58.1
	Dia. MP (Tokens)	63.8	**67.6**	64.3	61	63.6	62.5	62.4	57.9

Table 3: Full results for pre-trained and fine-tuned (FT) models. For each task, the best-performing model among those we pre-trained and fine-tuned (excluding the baseline) is shown in bold.

tokens from the prompts provided to `llamalogue` and the single ground-truth response generated by the teacher LLM. For the Teacher Confidence-based reward strategy, where ten teacher responses were used for confidence estimation, the total amounts to 9 903 146 tokens. In both cases, the overall token count remains well below the 100M-token limit specified by the BabyLM Challenge for the interaction track.[3]

3.4 Evaluation

Standard benchmarks For evaluation purposes, we rely on the BabyLM evaluation pipeline (Charpentier et al., 2025). As zero-shot evaluation, it includes minimal pairs tasks on the syntactic level (BLiMP, Warstadt et al., 2020) and on the semantic/world knowledge level (COMPS, Misra et al., 2023; EWoK, Ivanova et al., 2025; entity tracking, Kim and Schuster, 2023). Additionally, in further tasks, model probabilities/surprisal values are correlated with word-level age of acquisition (Chang and Bergen, 2022), cloze probabilities (De Varda et al., 2023), and preferences in morphological inflection for 'wug' words (Hofmann et al., 2025). Finally, the models are also evaluated through fine-tuning on a selection of tasks from GLUE and SuperGLUE (Wang et al., 2018, 2019).

Custom benchmarks To evaluate the models in a more holistic way, we include three additional minimal pair benchmarks. We (i) create a dialogue minimal pair set. As already described in Section 3.2, positive examples are created by simply matching parental utterances with children's answers, negative examples are sampled by matching the same parental utterances with unrelated child utterances. With this dataset, we aim to not only test the formal language skills of our models (as the BabyLM evaluations already do), but also their functional skills (Mahowald et al., 2024). Furthermore, we include (ii) Zorro (Huebner et al., 2021), a reduced version of BLiMP with a vocabulary restricted to words that occur in CHILDES, and (iii) the lexical decision dataset by Bunzeck and Zarrieß (2025), which contains word-level minimal pairs of words and non-words (e.g., *sending* and *monding*) as benchmarks that should be more tuned to the linguistic register found in our pretraining data.

4 Results

4.1 Base model evaluation

We evaluate our base model after being trained for 10 epochs. We compare `llamalogue` to the baseline model `babylm-interaction-baseline-simpo`[4] provided by the BabyLM organizers for the *interaction* track. Our model performs worse than this baseline model in almost every BabyLM evaluation task, except entity tracking (cf. Table 3). In comparison to other models submitted to the *strict-small* track, our model performs particularly worse on

[3]The code for DPO and PPO experiments can be found at these two Github repositories: https://github.com/fpadovani/communicative_baby_ppo and https://github.com/fpadovani/communicative_baby_dpo

[4]https://huggingface.co/BabyLM-community/babylm-interaction-baseline-simpo

BLiMP and AoA prediction, whereas scores for EWoK, COMPS, (Super)GLUE or the different wug tests are undercut by several other submissions. Therefore, `llamalogue` is not a generally bad language model, but its pretraining peculiarities have a non-straightforward effect on performance.

With respect to the custom benchmarks the results are more nuanced. For example, on the *dialogue minimal pairs* task, which aligns closely with the pre-training goal of `llamalogue`, it exhibits a clear advantage over the baseline comparison (63–64% vs. 57–58%). Our model also achieves a reasonable accuracy of 65.5% on *Zorro*. Nevertheless, it is clearly outperformed by the interactive baseline (77.7%) which was trained on the full BabyLM data. Our base model also falls behind the interactive baseline on the *lexical decision* task, performing quite far (40.3%) below chance level.

4.2 Fine-tuned models

4.2.1 BabyLM evaluations

Like the `llamalogue` base model, our fine-tuned models show overall lower performance on almost all of the zero-shot BabyLM Challenge tasks than the baseline model and the other models submitted to the interaction track. For BLiMP, all model variants score substantially below the baseline's 72.16%, with results clustering around chance level. The highest score is achieved by the Semantic Similarity model at 56.36%. Similar trends hold for BLiMP supplementary, where the gap to the baseline remains notable. Surprisingly, for entity tracking our models improve over the baseline of 28.06, with the best score (36.03) achieved by a model fine-tuned with OLMo Score. For EWoK, scores are near chance level, in accordance with the baseline model. Reading-based tasks (eye-tracking and self-paced reading) show much lower alignment with human patterns than the baselines. The Wug adjective and Wug past morphological generalization tasks yield near-zero or negative correlations across all models, far from the baseline model score of 38.5 for Wug adjective, underscoring persistent difficulty in capturing human-like morphological generalization. For AoA, after a closer look at metric computation, we find that only very few data points (1–5 words) are considered. This is due to an unpassed condition on the parameters of the fitted sigmoid function within AoA computation in the evaluation pipeline. Limited data points lead to either a score of zero or a strong negative correlation; hence, these results can be misleading. Overall, while entity tracking shows a modest improvement over the baseline, most linguistic and psycholinguistic tasks still reveal substantial gaps. The usefulness of our models for fine-tuning is not affected by reinforcement learning, indicated by (Super)GLUE scores that do not change drastically and also remain lower than for the baseline model (cf. also Appendix D).

4.2.2 DPO reward and custom evaluations

As shown in the right plot of Figure 1, the reward assigned to acceptable and unacceptable utterances begins to diverge early in the fine-tuning process. This separation is particularly pronounced in the case where the acceptable sentence is artificially generated by the LLM, suggesting a stronger initial reward signal and a more stark contrast between both continuations. Interestingly, this tendency is not confirmed by performance on *dialogue minimal pairs*. Although both DPO models improve upon the base model with regard to this measure, the effect of the synthetic data is rather low (increase of approximately 0.5%).

In contrast, the model fine-tuned on real caregiver–child interaction data scores approximately 4% higher than the base model and the model fine-tuned on artificially generated child utterances. This suggests that, although LLM-generated utterances may be more grammatical and exhibit greater syntactic and lexical variety than real data found in CHILDES, the model fine-tuned on synthetic data is less apt at predicting real minimal pairs derived from genuine interactions. The natural data is clearly superior to synthetic data when trying to optimize for this task. For Zorro, the naturalistic model maintains performance comparable to `llamalogue`, whereas the synthetic model shows slightly lower accuracy.

4.2.3 PPO reward and custom evaluations

During PPO fine-tuning, we observe occasional instability[5] in the training process. To ensure consistency in evaluation, we assess all models at the end of the first epoch, after a single full pass over the novel data. For the OLMo-based score, the training process shows a sharp reward decline before completing the first full epoch. Therefore, we select an earlier checkpoint (5000 steps) for evaluation, under the assumption that these 5000 steps still

[5] Including abrupt drops in reward and unexpected script crashes before completion.

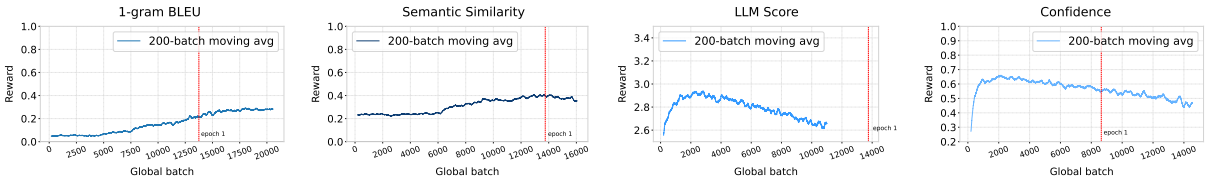


Figure 2: Reward trends over training steps for four reward metrics: 1-gram BLEU, Semantic similarity, LLM score, and Teacher Confidence-based. Vertical line marks the end of epoch 1. For LLM score and confidence, the y-axis range has been restricted to enhance visibility of trends, and does not represent the full possible reward scale.

provide a meaningful degree of fine-tuning before instability occurs.

1-gram BLEU Reward The reward starts off very low and remains low for a substantial number of steps before beginning to increase steadily. Given that this is a unigram-based metric focused on token overlap between the generated utterance and a reference, a slow and gradual increase is actually desirable, a sharp rise could lead the model to simply replicate the caregiver's utterances. For *Zorro*, this model achieves the lowest score among all those evaluated, and it also ranks among the least accurate models on the *dialogue minimal pairs*. Although it is a word-based metric, no further improvements on the *lexical decision* data can be reported.

Semantic Similarity Reward The reward increases gradually during training, similarly to what is observed for BLEU. However, the overall improvement across training steps is modest, and the reward values remain relatively low. On *Zorro*, the model's accuracy stays roughly at the level of `llamalogue`. Additionally, performance on the *dialogue minimal pairs* shows a slight decline of a few percentage points compared to the pre-trained model. The score on the *lexical decision* task is the lowest observed among all the fine-tuned models.

LLM-generated Reward Here, the reward increases during the very early phase of fine-tuning, although only by approximately 0.5 on a scale ranging from 0 to 5. This limited growth indicates that the OLMo model used to assign the reward rarely utilized the full range of available values. In particular, scores of 0 or 5 were almost never assigned to generated utterances. Starting from around step 3000, the reward begins to decline steadily. The model evaluated at checkpoint-5000 maintains a relatively strong performance on *Zorro*. However, similar to the previous two PPO models, there is a decrease in accuracy on the *dialogue MP* task compared to `llamalogue`. Zorro and *lexical decision* scores stay roughly equivalent to the base model.

Teacher Confidence-based Reward The initial reward being around 0.2 means the `llamalogue` reply was already above the median among the teacher's ten candidates. At the start of fine-tuning the reward increased quickly, probably due to the initially small KL coefficient value. During fine-tuning, the reward rose to around 0.6, meaning the fine-tuned model beats roughly eight of the teacher's alternatives. After epoch one, the reward curve had a slight dip to around 0.5. On the *lexical decision* task, the model is roughly on par with `llamalogue`, but lower on *Zorro* and (slightly) *dialogue MP*.

5 Discussion and Conclusion

How can these slightly underwhelming results be explained? First, we need to emphasize that our dialogue-only models, trained on child-directed and child speech, are exposed to a smaller vocabulary (Snow and Ferguson, 1977) and simpler structures (Genovese et al., 2020) than found in adult speech (although complex structures are occasionally found in CDS, they are rare, cf. Cameron-Faulkner et al., 2003). As the benchmarks included in BabyLM target broader lexical and syntactic variation in the input, there is a slight mismatch between our data and the evaluation data. The accuracy on lexically restricted Zorro, for example, is much higher than the one reported for BLiMP. More generally speaking, these results also align with previous findings on other models trained on CDS only (cf. Padovani et al., 2025; Bunzeck et al., 2025). Where our models excel is the domain of dialogue minimal pairs. There, they outperform the base model by a margin of 10%. While it is not overly surprising that our model masters a task that aligns 100% with its pre-training goal and the shape of its data, learning dialogue coherence is still far from easy. Judging contingency and coherence without lexical

overlap requires a different kind of linguistic knowledge than syntactic phenomena like island effects – exactly the kind of knowledge our model picks up.

With respect to the performance of our fine-tuned models, it is important to note that our results align with previous studies (Liu and Fourtassi, 2025; Stöpler et al., 2025), which all found no significant improvements on grammatical or similar benchmarks after interaction-driven fine-tuning. Such fine-tuning with a specific, pragmatics- or communication-based goal in mind has so far only shown to improve performance on benchmarks that also test for this goal. Our DPO fine-tuning, which directly optimizes preference for correct answers, does have a positive effect on the model preferring such answers from a held-out test set. In contrast, more generalized optimization for communicatively appropriate generations with PPO does not have this effect. It remains open to further inquiry whether our scoring methods might be too abstract. After all, they are only indirectly aligned with all the different evaluation measures we want to optimize for (correct grammar, world knowledge, approximation of human reading behaviour, AoA estimation, etc.). Also, if the one, singular answer that we compare with our generation in all PPO training regimens is too distant to the generated answer (semantically, pragmatically, lexically, etc.), then the provided training signal might steer the model's weights into incorrect directions or leads to it getting stuck in local optima (exemplified by the non-monotonic reward trends).

Finally, as the differences between DPO with naturally occurring and synthetically generated answers are quite large for the dialogue MP performance, this hints towards a shortcoming of current LLMs: despite generating language that superficially resembles CDS being easy, generating authentic interactions is actually hard. For example, Feng et al. (2024) generate synthetic dialogues which differ tremendously from real caretaker-child interactions – the utterances are not fragmentary, highly verbose and complex. Räsänen and Kocharov (2024) train a CDS model from scratch, which approximates many statistical tendencies of CDS, but often generates nonsensical or ungrammatical utterances. While our model did not perform well on the general BabyLM benchmarks, a first qualitative inspection of its generative capabilities showed that it can actually continue dialogue in a plausible-looking way. Here, further experimentation with dialogue-based models is clearly needed.

Limitations

This study has several limitations that should be acknowledged. First – as previously discussed – the training data is narrowly focused on child-directed and child speech, which, while intentional for our research goals, constrains the model's lexical diversity and syntactic variety. This domain-specific bias limits its generalization to broader linguistic contexts, as evidenced by weaker performance on benchmarks that target a wider range of grammatical phenomena such as BLiMP. The incorporation of adult–adult dialogue into our training regimen might be a promising direction for future research. However, our primary objective in this study was to optimize the child component's conversational turns in dialogic interactions with caregivers, while testing if this also enhances secondary objectives like semantic relevance, common-sense reasoning, and linguistic competence. In child language development, these abilities emerge through interleaved phases/periods characterized by imitation and strong reliance (exploitation) on parental input, and others dominated by exploration of self-generated abilities and emergent capacities. Transposed to the context of a reward function guiding model competencies over time, this developmental dynamic could, for example, suggest the use of a curriculum-based reward schedule across fine-tuning steps. Such a schedule could involve intensifying the reward signal during certain stages and attenuating it during others, or alternatively optimizing different aspects of verbal production at distinct developmental phases of the model. Notably, our study did not incorporate such a curriculum in the reward design, which may have limited the effectiveness of the PPO fine-tuning. It would be interesting for future work to explore this direction and assess whether exploration/exploitation reward patterns inspired by human developmental trends could yield greater benefits for model fine-tuning.

Furthermore, our fine-tuning phases with DPO and PPO were conducted without a previous extensive hyperparameter search. As a result, the (in-)effectiveness of our proposed reward functions and their learning dynamics remain open for further exploration. Importantly, our project is intended as a pilot study. While we put more emphasis on the comparison of and experimentation with a broad variety of studies, future work should place greater emphasis on systematically identifying the optimal hyperparameters for each reward function prior to

training, thereby ensuring that observed effects can be more confidently attributed to the reward design itself rather than possibly suboptimal fine-tuning setups.

Supplementary Materials

In addition to being conveniently available on Huggingface and GitHub, the long-term accessibility of the datasets, models, and code for the DPO and PPO experiments is ensured via a data publication on Zenodo: https://doi.org/10.5281/zenodo.17253651

Acknowledgments

BB, MA, OM, HB and SZ acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): CRC 1646/1 2024 – 512393437 projects A02, A05, and B02. FP and AB were supported by the Talent Programme of the Dutch Research Council (grant VI.Vidi.221C.009).

References

Heike Behrens. 2021. Constructivist approaches to first language acquisition. *Journal of Child Language*, 48(5):959–983.

Robert C. Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive Science*, 35(7):1207–1242.

Bastian Bunzeck and Holger Diessel. 2025. The richness of the stimulus: Constructional variation and development in child-directed speech. *First Language*, 45(2):152–176.

Bastian Bunzeck, Daniel Duran, and Sina Zarrieß. 2025. Do construction distributions shape formal language learning in German BabyLMs? In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 169–186, Vienna, Austria. ACL.

Bastian Bunzeck and Sina Zarrieß. 2024. Fifty shapes of BLiMP: Syntactic learning curves in language models are not uniform, but sometimes unruly. In *Proceedings of the 2024 CLASP Conference on Multimodality and Interaction in Language Learning*, pages 39–55, Gothenburg, Sweden. ACL.

Bastian Bunzeck and Sina Zarrieß. 2025. Subword models struggle with word learning, but surprisal hides it. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 286–300, Vienna, Austria. ACL.

Thea Cameron-Faulkner, Elena Lieven, and Michael Tomasello. 2003. A construction based analysis of child directed speech. *Cognitive Science*, 27(6):843–873.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM turns 3: Call for papers for the 2025 BabyLM workshop. *Preprint*, arXiv:2502.10645.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts bert. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 210–224, Singapore. ACL.

Noam Chomsky. 1986. *Knowledge of Language. Its Nature, Origin and Use.* Praeger, New York, NY, USA.

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. 2022. The grammar-learning trajectories of neural language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8281–8297, Dublin, Ireland. ACL.

Andrea Gregor De Varda, Marco Marelli, and Simona Amenta. 2023. Cloze probability, predictability ratings, and computational estimates for 205 English sentences, aligned with existing EEG and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Evelina Fedorenko, Steven T. Piantadosi, and Edward A. F. Gibson. 2024. Language is primarily a tool for communication rather than thought. *Nature*, 630(8017):575–586.

Steven Y. Feng, Noah Goodman, and Michael Frank. 2024. Is child-directed speech effective training data for language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22055–22071, Miami, Florida, USA. ACL.

Giuliana Genovese, Maria Spinelli, Leonor J. Romero Lauro, Tiziana Aureli, Giulia Castelletti, and Mirco Fasolo. 2020. Infant-directed speech as a simplified but not simple register: A longitudinal study of lexical and syntactic features. *Journal of Child Language*, 47(1):22–44.

Alison Gopnik. 2020. Childhood as a solution to explore–exploit tensions. *Philosophical Transactions of the Royal Society B*, 375(1803):20190502.

Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. ACL.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. ACL.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of World Knowledge (EWoK): A cognition-inspired framework for evaluating basic world knowledge in language models. *Preprint*, arXiv:2405.09605.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 3835–3855, Toronto, Canada. ACL.

Seokyung Kim and Stephanie M. Carlson. 2024. Understanding explore–exploit dynamics in child development: Current insights and future directions. *Frontiers in Developmental Psychology*, 2:1467880.

Stephen C. Levinson. 2025. *The Interaction Engine: Language in Social Life and Human Evolution*. Cambridge University Press, Cambridge, UK.

Jing Liu and Abdellah Fourtassi. 2025. Benchmarking LLMs for mimicking child-caregiver language in interaction. *Preprint*, arXiv:2412.09318.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. *Preprint*, arXiv:2505.24864.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021. Probing across time: What does RoBERTa know and when? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 820–842, Punta Cana, Dominican Republic. ACL.

Ziqiao Ma, Zekun Wang, and Joyce Chai. 2025. Babysit a language model from scratch: Interactive language learning by trials and demonstrations. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 991–1010, Albuquerque, NM, USA. ACL.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*, 3rd edition. Lawrence Erlbaum, Mahwah, NJ, USA.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. ACL.

Mitja Nikolaus and Abdellah Fourtassi. 2025. Modeling children's grammar learning via caregiver feedback in natural conversations. *Preprint*, OSF:e6kv3_v1.

Kate Nussenbaum and Catherine A. Hartley. 2019. Reinforcement learning across development: What insights can we draw from a decade of research? *Developmental Cognitive Neuroscience*, 40:100733.

Jeffrey Olmo, Jared Wilson, Max Forsey, Bryce Hepner, Thomas Vincent Howe, and David Wingate. 2025. Features that make a difference: Leveraging gradients for improved dictionary learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7609–7619, Albuquerque, NM, USA. ACL.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran.

Francesca Padovani, Jaap Jumelet, Yevgen Matusevych, and Arianna Bisazza. 2025. Child-directed language does not consistently boost syntax learning in language models. *Preprint*, arXiv:2505.23689.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA. ACL.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran.

Okko Räsänen and Daniil Kocharov. 2024. Age-dependent analysis and stochastic generation of child-directed speech. In *Proceedings of the 46th Annual Meeting of the Cognitive Science Society*, pages 5102–5108, Rotterdam, The Netherlands.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, Hong Kong, China. ACL.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is more: Pre-training cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Catherine E. Snow and Charles A. Ferguson, editors. 1977. *Talking to Children: Language Input and Acquisition*. Cambridge University Press, Cambridge, MA, USA.

Lennart Stöpler, Rufat Asadli, Mitja Nikolaus, Ryan Cotterell, and Alex Warstadt. 2025. Towards developmentally plausible rewards: Communicative success as a learning signal for interactive language models. *Preprint*, arXiv:2505.05970.

Jean-Loup Tastet and Inar Timiryasov. 2024. Baby-Llama-2: Ensemble-distilled models consistently outperform teachers with limited data. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 292–301, Miami, FL, USA. ACL.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Cambridge, MA, USA.

Michael Tomasello. 2005. Beyond formalities: The case of language acquisition. *The Linguistic Review*, 22(2-4):183–197.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformer Reinforcement Learning. https://github.com/huggingface/trl.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran, Vancouver, Canada.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. ACL.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–6, Singapore. ACL.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Aditya Yedetore, Tal Linzen, Robert Frank, and R. Thomas McCoy. 2023. How poor is the stimulus? Evaluating hierarchical generalization in neural networks trained on child-directed speech. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 9370–9393, Toronto, Canada. ACL.

Xingmeng Zhao, Tongnian Wang, Sheri Osborn, and Anthony Rios. 2023. BabyStories: Can reinforcement learning teach baby language models to write better

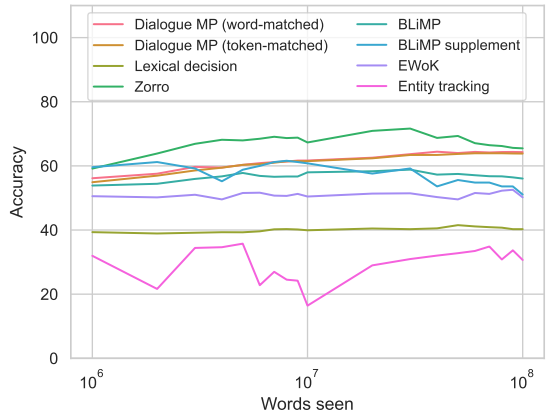


Figure 3: Learning trajectories for our base model across pre-training for 10 epochs. Note that the x-axis is log-scaled to make the very early training dynamics more visible.

stories? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 158–169, Singapore. ACL.

A Learning trajectories across pretraining

To trace the learning process of `llamalogue`, we continually evaluate it during pretraining. We benchmark ten checkpoints across the first epoch (so after each 1M token set has been seen by the model once, until 10M tokens are reached) and then nine further checkpoints over the remaining nine epochs. We visualize the development of performance on eight different minimal pair sets in Figure 3.

The worst performance can be observed for the entity tracking evaluation – performance does not stabilize at all and oscillates between 20–40%, which means that our model actively disprefers correct continuations. The same goes for the lexical decision data, where our model consistently scores around 40%. Performance on EWoK stays around the chance baseline as well. Interestingly, our model surpasses 60% on the BLiMP supplement data around approximately 7M tokens, after which performance deteriorates again. Similarly, BLiMP performance increases slightly early on, but then also stabilizes at a low level. Accuracy scores on Zorro, the scaled-down derivative of BLiMP that only contains words also occurring in CHILDES, are generally higher and improve until the third epoch of training, after which they deteriorate again. The only stable, monotonically improving learning trajectory can be observed for our dialogue minimal pairs. This, however, is not overly surprising, as this testing paradigm aligns closely with the pretraining goal of `llamalogue`. Viewed in conjunction with our general results, these learning trajectories further corroborate the fact that the general BabyLM evaluation measures are not very suitable for our models, as the decreasing learning trajectories hint towards our models not being undertrained and because comparable studies of learning dynamics overwhelmingly report power-law like curves (cf. Huebner et al., 2021; Liu et al., 2021; Choshen et al., 2022; Bunzeck and Zarrieß, 2024; Padovani et al., 2025).

B DPO Datasets

Table 4 shows a sample of sentences from the dataset we used to fine-tune the model with DPO. The appropriate and random sentences are matched in terms of token length, and both come from the distribution of sentences actually observed in CHILDES.

Table 5 was also used to fine-tune `llamalogue` with DPO. In contrast to the previous case, the appropriate sentences here are synthetic, artificially generated by Llama-3.2-3B, and their length is not matched to that of the random counterparts.

C PPO Reference Child Responses

In Table 6, we show a sample of 3 prompts used during fine-tuning and the 10 ground truth answers generated by Llama-3.2-3B when it is asked to simulate a child responding to a caregiver's sentence, using the prompt shown in detail in Table 1.

D (Super)GLUE results

We report the results for the SuperGLUE tasks in Table 7. Here, we can generally report that fine-tuning with DPO and PPO has only very little effect on our models' advantages for further fine-tuning. In comparison to the baseline model trained on the whole BabyLM corpus, they are generally worse base models for fine-tuning on (Super)GLUE.

Prompt (MOT)	Appropriate (CHI)	Random (CHI)
what is that ?	it looks like a gun .	you do it like that .
pull the string .	and where do they hook it ?	do you know what it was ?
I think they just throw it on the side .	you know what Mom ?	I get this hole .
what are you playing with huh toys ?	there's a dog .	there's the sports .
the bottom ones come off .	want to know what ?	we stole the brush .
can you say that ?	okay the hungry hungry caterpillar .	yeah I want that too .
what is it ?	a baby caterpillar !	I'm just pretending .
what is it ?	I don't .. know !	put my dress down .

Table 4: Examples of naturalistic DPO dialogue pairs. Each row shows a caregiver's utterance (MOT) as prompt, with corresponding appropriate and random child responses (CHI) used for preference training.

Prompt (MOT)	Appropriate (CHI)	Random (CHI)
what is that ?	it is a ball.	you do it like that .
pull the string .	i can do it!	do you know what it was ?
I think they just throw it on the side .	the garbage truck.	I get this hole .
what are you playing with huh toys ?	i'm playing with this toy, it's a nice toy.	there's the sports .
the bottom ones come off .	the bottom ones come off .	we stole the brush .
can you say that ?	yes, mother can say that.	yeah I want that too .
what is it ?	i don't know.	I'm just pretending .
what is it ?	a big tree.	put my dress down .

Table 5: Examples of synthetic DPO dialogue pairs. Each row shows a caregiver's utterance (MOT) as prompt, with corresponding appropriate child responses (CHI) generated using the Llama-3.2-3B teacher model, and random responses from naturalistic mismatches.

Prompt (MOT)	Generated child-like responses (LLaMA-3B)
I was wondering who told you your neck needed washing.	1. It was a frog. 2. The neck was washed, so I asked the dog to lick it. 3. No one told me. 4. I don't know, mom. 5. I don't know, but I think I'll wash it now. 6. I don't know, I just felt it needed washing. 7. I think it was you. 8. It was the person in the mirror. 9. Mommy told me to wash my neck. 10. Mother, I was wondering who told you my neck needed washing.
I think they're all friends together, aren't they?	1. Yes, mother, I think so. 2. I think they're all friends together, aren't they? 3. I think they're all friends together, aren't they? 4. Yes, they are. 5. Yes, mother, they are. 6. Yes, I think so. 7. Yes, they are, Mom. 8. I think they're all friends together, aren't they? 9. Yes, I think so, aren't they? 10. Yes, they are friends.
You're a nice girl.	1. you're a nice girl too. 2. I'm sorry, I'm not a nice girl, I'm a bad girl. 3. Yes, I am. 4. I am a nice girl. 5. thank you, mom. 6. yes, I am a nice girl. 7. thank you mother. 8. I'm nice too. 9. thank you, mom. 10. yes I am.

Table 6: Caregiver prompts and ten possible child-like answers generated by the Llama-3.2-3B model.

		DPO		PPO				
Task	**llamalogue**	**Natural.**	**Synth.**	**Bleu**	**SemSim**	**LM Score**	**Conf.**	**Baseline**
BoolQ (acc)	64.04	64.04	64.04	64.04	64.04	64.04	64.04	68.38
MNLI (acc)	35.17	35.17	34.92	34.60	34.82	35.23	34.92	61.04
MRPC (F1)	80.95	80.95	80.95	81.31	80.95	81.31	80.95	83.61
QQP (F1)	10.28	10.28	10.17	5.55	11.13	10.37	11.17	71.82
RTE (acc)	53.24	52.52	53.24	53.24	54.68	51.80	53.24	61.15
MultiRC (acc)	57.55	57.55	57.55	57.55	57.55	57.55	57.55	65.92
WSC (acc)	61.54	61.54	61.54	61.54	61.54	61.54	61.54	63.46

Table 7: SuperGLUE results.

CLASS-IT: Conversational and Lecture-Aligned Small-Scale Instruction Tuning for BabyLMs

Luca Capone[1*†] **and Alessandro Bondielli**[1,2†] **and Alessandro Lenci**[1†]

[1]CoLing Lab, Department of Philology, Literature and Linguistics, University of Pisa
[2]Department of Computer Science, University of Pisa

luca.capone@fileli.unipi.it, {alessandro.bondielli, alessandro.lenci}@unipi.it

Abstract

This work investigates whether small-scale LMs can benefit from instruction tuning. We compare conversational and question-answering instruction tuning datasets, applied either in a merged or sequential curriculum, using decoder-only models with 100M and 140M parameters. Evaluation spans both fine-tuning (SuperGLUE) and zero-shot (BLiMP, EWoK, WUGs, entity tracking, and psycholinguistic correlation) settings. Results show that instruction tuning yields small but consistent gains in fine-tuning scenarios, with sequential curricula outperforming merged data; however, improvements do not consistently transfer to zero-shot tasks, suggesting a trade-off between interaction-focused adaptation and broad linguistic generalization. These results highlight both the potential and the constraints of adapting human-inspired learning strategies to low-resource LMs, and point toward hybrid, curriculum-based approaches for enhancing generalization under ecological training limits.

1 Introduction

The role of input data vis-à-vis innate biases has long dominated the debate on language acquisition. This is exemplified by arguments such as the poverty of the stimulus and the language of thought hypothesis (Chomsky, 1980; Fodor, 1975), which have emphasized the need for innate constraints governing the process of acquiring productive linguistic generalizations. In contrast, data-driven learning has always been a central tenet of connectionist theory, arguing that, given sufficient training, a large enough model can reproduce any regular behavioral pattern (Smolensky, 1988). One of the defining features of LMs is that performance relies on the training process. The development of model abilities clearly reflects learning, although the precise nature of this learning is not yet well understood. It remains uncertain whether abilities (or at least some of them) are truly emergent Wei et al. (2022), or whether this impression is an artifact of measurement, with capabilities in fact increasing more gradually (Schaeffer et al., 2023). Moreover, the type and order of training data can influence a model's ability to perform specific tasks (Soviany et al., 2022). Finally, particular training regimes, such as instruction tuning or reinforcement learning with human feedback (RLHF), can significantly enhance a model's capacity for user interaction, as well as its logical, inferential, and reasoning abilities. Despite relying on radically different mechanisms, LMs and humans share several key properties of learning: both improve with training over time, both are sensitive to the quality of instruction, and both benefit from interactive, feedback-driven training. These parallels suggest that current LMs approximate some aspects of human-like learning. However, the scale of resources required (both in terms of data and computation) remains orders of magnitude greater than what is needed for human learning, especially in children (Frank, 2023). Among these shared features, this paper focuses on **interaction**, a core component of human learning, particularly in childhood. We investigate whether an LM trained on ecologically valid input, comparable in scale to the linguistic exposure of a 10-year-old child, can benefit significantly from targeted instruction tuning. Specifically, we compare two types of instruction tuning datasets: one centered on conversational interactions and the other focused on question-answering tasks. The main research questions addressed in this study are:

- Can a BabyLM benefit from instruction tuning?
- Given the limited pre-training typical of BabyLMs, which type of instruction data is

*Corresponding author

†For the specific purposes of Italian Academy, Luca Capone is responsible for Sections 2, 3 and 4, Alessandro Bondielli is responsible for sections 5 and 6, Alessandro Lenci is responsible for sections 1 and 7.

Proceedings of the First BabyLM Workshop, pages 448–456

more effective: conversational or open-ended question-answering?

- Does a curriculum learning approach to instruction tuning provide significant benefits?

This paper is organized as follows. Section 2 reviews related work. Section 3 describes the datasets used for pre-training and instruction tuning. Section 4 presents the model architectures and details the training procedures, while Sections 5 and 6 present and analyze the results on the BabyLM Challenge tasks. Finally, Section 7 summarizes our findings and outlines directions for future research.

2 Related Works

While early language learning in children is often portrayed as remarkably precocious (McCormack and Hoerl, 2005; Gopnik, 2011; Dündar-Coecke et al., 2020), linguistic and psychological studies suggest that this view must be qualified. Many scholars acknowledge children's early communicative abilities, but argue that these are constrained to specific tasks and contexts, and do not necessarily reflect a fully developed understanding of language. For instance, although the intersubjective (i.e., social and communicative) function of linguistic signs becomes evident in children from an early age, their perspectival function, the ability to conceptualize experiences from multiple viewpoints, emerges more gradually (Vygotsky, 1987; Piaget, 2002; Tomasello, 2009).

Drawing on developmental psycholinguistic evidence (Berman and Slobin, 2013; Peterson and McCabe, 1987), Tomasello (2003) observed that many children up to the age of nine, despite producing fluent, age-appropriate speech, struggle to use sophisticated conjunctions (such as *because, indeed, although*, etc.) when required to do so. These conjunctions involve representing events from a logical–causal or antithetical perspective, which can pose significant challenges. At this stage, *and* remains the most frequently used connective, functioning in an undifferentiated way to express a wide range of semantic relations, even after more specific connectives have begun to appear in a child's speech. Similar limitations occur with other complex constructions: comprehension and voluntary use often do not match the apparent fluency of spontaneous speech. Berman and Slobin (2013) document the difficulties children face with narrative discourse, sometimes even up to age nine, when asked to describe a story depicted in a sequence of images. Children frequently struggle to produce coherent narratives that clearly indicate a beginning, progression, and conclusion. Tomasello (2003) attributes these challenges to the *plurifunctionality* of complex constructions, arguing that mastery of the perspectives they encode develops gradually over the course of the school years.

Building on this body of research, the present study investigates whether and to what extent interactive instruction can enhance the training of a BabyLMs. In particular, it examines whether formal, instruction-like input provides greater benefits than conversational data for fine-tuning LMs trained on limited, child-comparable linguistic exposure. To our knowledge, the two main attempts to interactively train BabyLMs using more pedagogically structured data are Baby's CoThought (Zhang et al., 2023) and Baby Stories (Zhao et al., 2023). However, both differ from the approach proposed in this work, albeit for different reasons. Zhang et al. (2023) build an educational dataset based on the BabyLM Challenge trainset, using GPT-3.5-Turbo. However, the dataset is used to train an encoder-only model through masked language modeling. Zhao et al. (2023), on the other hand, preserves an interactive setup by fine-tuning a decoder model using proximal policy optimization. Nonetheless, this training technique departs from the type of formal instruction we aim to address, as the model is simply optimized to prefer certain generations based on a reward model. In contrast, the instruction tuning proposed in this work more closely resembles the structured, formal education typically provided to children in school settings. The present work instead adopts an instruction fine-tuning approach, where models are explicitly trained to respond to questions about specific topics and to provide appropriate answers within conversational contexts.

3 Dataset

The dataset used for model pre-training is a curated subset of the data provided by the task organizers, which amounts to approximately 91 million words. The instruction tuning dataset includes processed Switchboard transcripts and augmented Simple Wikipedia texts, enhanced using the `LLaMA-3.2-3B-Instruct` model (Dubey et al., 2024).

3.1 Pretraining Dataset

The data supplied by the organizers (approximately 100 million words) underwent standard preprocessing. Special characters were removed, and all entries containing two words or fewer were discarded. Additional processing was applied to the Switchboard corpus, utterances from the same speaker were concatenated when they occurred in sequence, following standard dialogue normalization practices. Roughly 75 million words—drawn from CHILDES, Gutenberg, BNC and OpenSubtitles—were used exclusively for pre-training. An additional 16 million words (from Switchboard and Simple Wikipedia) overlap with the instruction tuning dataset, bringing the total pre-training corpus to approximately 91 million words.

3.2 Instruction-Tuning Dataset

The instruction tuning dataset consists of two sections: a **conversational component** based on the Switchboard corpus and an **instructional component** based on Simple Wikipedia. For the conversational section, the Switchboard data were adapted to meet the requirements of instruction tuning training task. Consecutive utterances from the same speaker were merged to ensure a consistent alternation between speakers' turns (e.g., A, B, A, B). The dialogues were then segmented into prompt–reply pairs using a sliding window approach with the following schema: (A1, B1), (B1, A2), (A2, B2). The resulting dataset contains 38,802 items and approximately 1.3 million words (excluding prompt-reply duplicates). For the instructional section, Simple Wikipedia data were augmented using `LLaMA-3.2-3B-Instruct` (Dubey et al., 2024). For each article text, three question–answer pairs were generated using structured generation with *outlines*[1] and the following prompt:

> Based on the following text, generate 3 questions and detailed, informative answers. Each answer should be easy for a young person to understand and at least 2–3 sentences long. Explain things in simple language, with clear and friendly sentences. Avoid short or vague replies and give enough detail so a kid can learn something new.

The generated data significantly exceeds the word limit imposed by the challenge. In this work we use only a representative portion of the whole dataset (`colinglab/CLASS_IT`). The full dataset will be released in the future following appropriate validation. The subset used in this study contains 8.7 million words, keeping the total—along with the 91 million pre-training words (which already include Switchboard and Simple Wikipedia texts)—within the 100-million-word limit. The augmented Simple Wikipedia dataset includes 97,697 items, totalling 18 million words (Figure 1).

[1] https://dottxt-ai.github.io/outlines/latest/#acknowledgements

Hyperparameter	llama140M	llama100M
Vocab size	32,000	16,384
Max length	6,144	6,000
Hidden size	704	512
Attention heads	11	8
Layers	12	20
Trainable parameters	140,231,872	100,684,288

Table 1: Model architectures

Hyperparameter	Pretrain	Instr. tuning
Initial LR	2e-4	2e-5
Batch size	8	8
Maximum epochs	8	10
LR scheduler	linear	cosine w/ restarts
Warm-up steps	5,000	500

Table 2: Training parameters

4 Models and training

We trained two models (Table 1), both based on decoder-only, LLaMA-style architectures with large maximum sequence lengths to accommodate the long texts present in the instruction tuning dataset. The first model has 140 million parameters, featuring a larger hidden size and vocabulary size. Following its training, we developed a second model with approximately 100 million parameters, using a reduced hidden size and a vocabulary size comparable to baseline models.

Both models followed the same pre-training procedure. The tokenizer was trained on the entire available corpus before pre-training began. Models were then pre-trained for 8 epochs using the parameters in Table 2, processing a total of approximately 728 million words.

Instruction tuning use a different set of hyperpa-

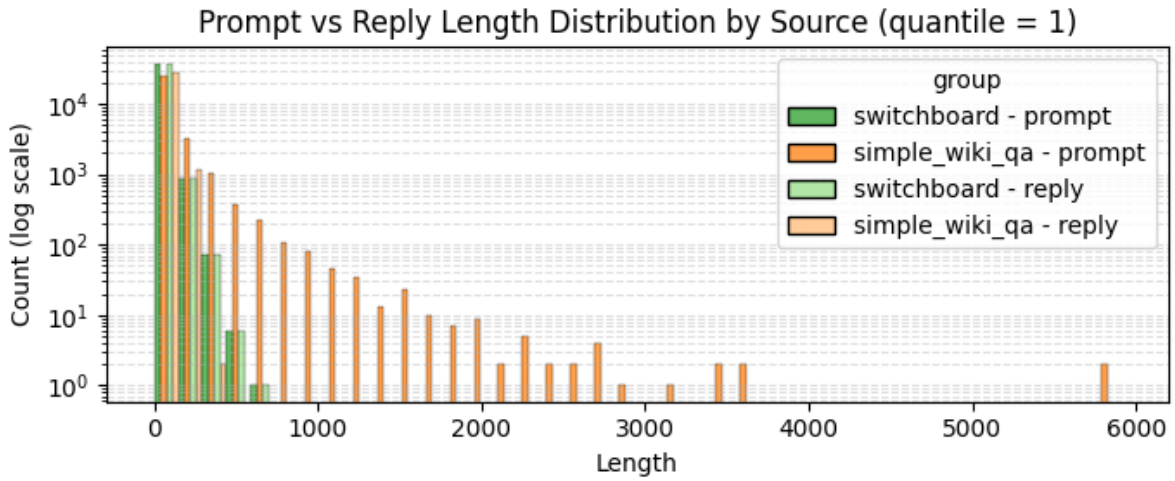


Figure 1

rameters from pre-training, but the same instruction tuning configuration was applied to both models (see Table 2). Each model was fine-tuned for 10 epochs, processing an additional 180 million words. In total, each model processed around 908 million words across both pre-training and instruction tuning. All datasets were split 90/10 into training and validation sets, with only the training portion contributing to parameter updates. Consequently, roughly 90% of the 908 million words (about 817 million) directly influenced model weight updates. We used the same token-level cross-entropy loss used for pre-training. However, for instruction tuning, we compute the loss only on target tokens, e.g. the answer tokens in a question-answer data point.

We adopted two strategies for instruction tuning: **merged** and **sequential**. In the merged strategy, augmented Simple Wikipedia data were shuffled together with Switchboard data, mixing conversational and instructional items. This produced the `it_merged` models (see Figure 2). In the sequential strategy, the two datasets were used in succession, resulting in two variants: `it_switch_wiki` and `it_wiki_switch`, depending on the order in which the pre-trained model was exposed to the instruction tuning datasets. This approach was designed to test whether keeping the tasks separate—and whether the order of exposure—provides measurable benefits to model performance.

5 Evaluation and Results

To evaluate our models, we used the official data provided by the challenge organizers. The evaluation is distinguished between a fine-tuning evaluation and a zero-shot evaluation.

Fine-Tuning Evaluation. In the fine-tuning evaluation, the models are fine-tuned and evaluated in the (Super)GLUE (Sarlin et al., 2020) tasks. We leave all the default parameters unchanged during training on each task. Note that the fine-tuning dataset is composed of a randomly sampled 10k portion of the original training set for the task. Models are evaluated on the test set. Figure 2 shows the result of our models (both pre-trained and instruction-tuned) and the baselines (`bl_gpt2-100M`, `bl_gptbertmixed-100M`, `bl_simpo`, shown in blue).

We observe that our models are generally competitive with the baselines, albeit surpassing them only for some configurations in the WSC task. As for the model size, the 140 million parameter models are generally better than the 100 million ones. However, the only tasks where the difference is noticeable are QQP and MNLI. Here, the 100 million models are markedly worse than their 140 million siblings, and both are markedly worse than the baselines. As for the instruction fine-tuning, it seems relatively beneficial. We see in fact that in all cases there is at least an instruction-tuned model better than the pre-trained one. However, differences are small and inconsistent across tasks, that is, there is no instruction tuning configuration that systematically leads to better results.

Since we have multiple tasks and models, we needed a way to compare performance globally. To achieve this, we standardized the results by computing z-scores for each model on each task, which express how many standard deviations above or

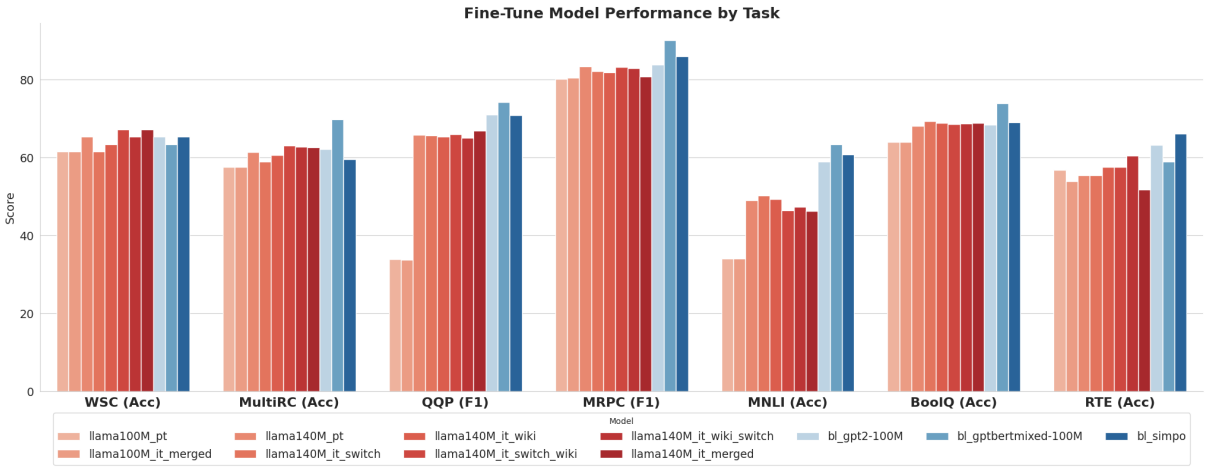


Figure 2: Results of fine-tuned models on (Super)Glue tasks.

below the task mean a model's score lies. We then averaged these z-scores across tasks to obtain a single global index per model. This index reflects the overall relative standing of a model compared to others, rather than absolute task performance, and allows fair comparison across heterogeneous metrics. Specifically, we plot them including median, Inter-Quartile Ranges (IQR), and outliers. Results are in Figure 3.

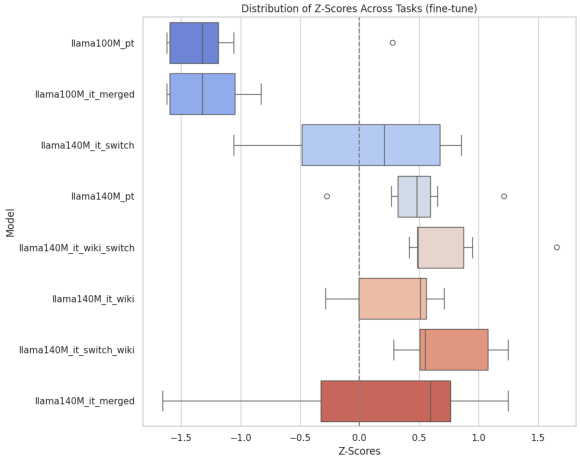


Figure 3: Median, Inter-Quartile Ranges (IQR), and outliers for z-scores of each model in the **fine-tuning** evaluation.

We observe that the 100 million models, both pre-trained and instruction-tuned, have negative z-score medians, while all the 140 million variants are on the positive side of the plot, showcasing that differences between the smaller and larger models appear to be significant. Regarding the differences between pre-training only and instruction tuning, we notice some interesting aspects. The model with the highest median is the merged instruction-tuned variant trained on a mixture of the datasets. However, both models trained on the two dataset sequentially, regardless of the order, have a very similar median score, but a much smaller IQR, and are the only two models with all z-scores above zero. Variants trained only on one of the datasets perform worse, and on par with the pre-trained only model, with the one trained just on Switchboard performing worst.

Zero-shot Evaluation. In the zero-shot scenario, models are evaluated using log-probabilities of sequences and/or words to obtain either model predictions or compute correlations with human data. The zero-shot evaluation is conducted on the following datasets: BLiMP (Warstadt et al., 2020) and EWoK (Ivanova et al., 2024) are standard minimal pairs datasets that test linguistic and world knowledge of LLMs and were included also in previous years' evaluations; a WUGs task designed to understand abilities in adjective nominalization (Hofmann et al., 2025); an entity tracking task on data from (Kim and Schuster, 2023); a correlation evaluation where cloze probability, predictability ratings, and computational estimates are compared against EEG and human reading time data (de Varda et al., 2024).

Results are reported in Figure 4. For the accuracy-based tasks, we do not observe striking differences between pre-trained and instruction-tuned models, similarly to what seen in the fine-tuning evaluation. However, here we also do not observe large differences also between the 100 million and 140 million variants. Our models seem to vastly outperform baselines on the WUGs task, but are worse on the Entity Tracking task, on which all

models including baselines seem to struggle. As for the Change in R^2 based tasks, we observe some surprising results: The 100 million model variants are vastly superior to both the 140 million models and the baselines, which score almost zero with the exception of the GPT-BERT mixed model. We compute z-scores distribution also in this case, and report them in Figure 5.

For the zero-shot evaluation the z-score distribution is radically different. No model has all z-scores above zero, and only two of them has a median z-score above zero. The two 100 million variants are among the best performing models, albeit this could be attributed to the vast differences between them and all the other models on the R^2-based tasks. The best performing model is an instruction-tuned variant, specifically the one trained only on Simple Wikipedia. However, no clear trend in favour or against instruction tuning emerge from the plot.

In order to further examine the performances on a broader level, we also plot the z-score distribution including both zero-shot and fine-tuning evaluations. Results are shown in Figure 6. It highlights the fact that, overall, the larger models seem to perform better. As for the impact of instruction tuning, we can highlight three aspects. First, we see that the best overall model is an instruction-tuned one. However, we cannot extrapolate a clear trend in favour of instruction tuning. Second, we observe that tuning the model sequentially on different datasets is consistently better than doing so on a mixture of the datasets. The order of the instruction tuning task seems less relevant, albeit we see that tuning first on conversational data (Switchboard) and then on question answering (Simple Wiki) seem to yield better results. This however may be affected by the difference in size between the datasets. In fact, we see that the model trained only on question answering performs better than the one trained subsequently on conversations.

6 Discussion

Our experiments provide some interesting insights about small-scale instruction tuning models trained on ecological amounts of data.

First, we see that instruction tuning appears to be somewhat beneficial, especially if the model is further fine-tuned on specific tasks; the same improvement are not as apparent on the zero-shot evaluation. We can hypothesize that the instruction tuning stage varies the models' internal distribution to a higher degree, especially at this scale, thus affecting the performances on zero-shot tasks, where the encoding of grammar rules (BLiMP, WUGs) or specific facts (EWoK) is more relevant than conversational and/or generative performances, which are not tested here. The instruction tuned models may be biased to learn to solve a specific task, in our case following a conversation or answering factual questions, thus losing their generalization abilities on just language. This latter aspect is quite interesting in the context of BabyLMs, as larger models have been shown to not suffer from similar issues (Miliani et al., 2025). Further evidence for this effect can be seen in the fact that instruction-tuned models have seen more data than pre-trained ones, yet do not consistently outperform them.

Second, we observe that among our models, smaller ones are consistently better than larger ones at correlating with human data, with the pre-trained model being slightly better; only one baseline model, with a different architecture but of the same size as ours, achieve comparable results. This is in line with previous literature where smaller models often correlate better with human psychometric data (De Varda and Marelli, 2023; Oh and Schuler, 2023).

Finally, we see that all our models achieve relative consistent performance on a single-task basis, while being very inconsistent across tasks and evaluation methods; the same happens with baselines. This suggests that the constraints posed by the challenge itself, namely the amount of data and training compute allowed for a training run, may limit the generalization capability of decoder-only style models without additional modifications.

7 Conclusion and future work

This study examined whether BabyLM-scale models (trained on ecologically realistic amounts of linguistic input) can benefit from instruction tuning, and how different forms and orders of data affect their performance. Our findings show that instruction tuning yields modest but measurable gains in fine-tuning scenarios, particularly when conversational and question–answering datasets are presented sequentially rather than merged. However, these benefits do not translate consistently to zero-shot evaluations, suggesting that, at this scale, instruction tuning may bias models toward narrow interactional behaviors at the expense of broader

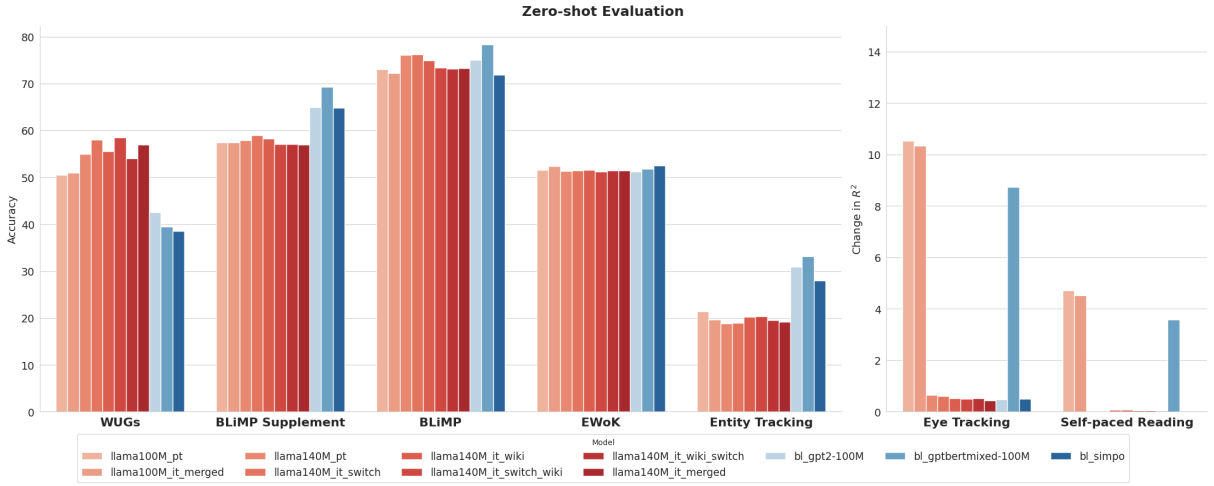


Figure 4: Results of the zero-shot evaluation. Tasks measured with accuracy are reported in the left bar chart; tasks measured with change in R^2 are reported in the bar chart on the right.

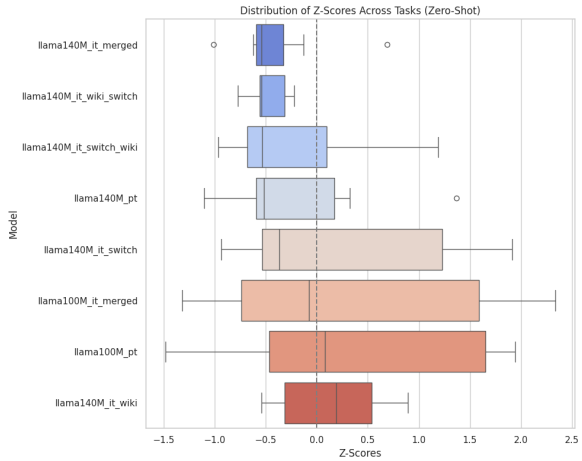


Figure 5: Median, Inter-Quartile Ranges (IQR), and outliers for z-scores of each model in the **zero-shot** evaluation.

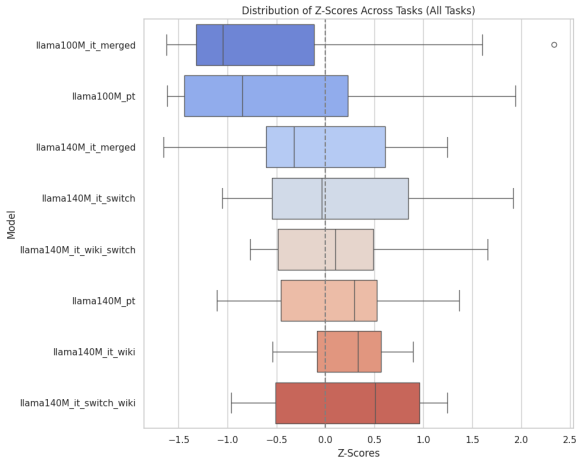


Figure 6: Median, Inter-Quartile Ranges (IQR), and outliers for z-scores of each model including both zero-shot and fine-tuning evaluations.

linguistic generalization.

A further limitation lies in how models are evaluated in the Challenge. In fact, in the fine-tuning evaluation most tasks are actually classification tasks, on which masked LMs may prove more reliable; in the zero-shot task, the vast majority of evaluations are conducted using log-likelihood as proxy for model choices. While this choice is valid in the context of the challenge, to accomodate the largest possible number of architectures and simplify the evaluation process, we can argue that model performances, especially when considering conversational instruction tuning, may be undermined by the evaluation criteria. Moreover, the chosen conversational portion of the instruction tuning dataset may limit the performances of the model: while the Switchboard corpus offers a structured and well-annotated resource, it represents a restricted register of spoken English and lacks much of the contextual diversity found in everyday interaction. More ecologically valid conversational data, spanning a wider range of speakers, settings, and discourse types, would provide a richer foundation for model adaptation and a stronger basis for subsequent instructional fine-tuning, potentially improving both interactive competence and generalization. Notably, smaller models exhibited stronger correlations with human psycholinguistic data, echoing prior observations that reduced capacity can sometimes yield representations more aligned with human processing patterns. Overall, the results highlight both the promise and the limi-

tations of adapting human-inspired learning strategies to small-scale LMs: interaction helps, but the gains are context-dependent, and generalization remains challenging under strict data and compute constraints. Future work should explore hybrid approaches that combine instruction tuning with targeted multi-task or curriculum learning, investigate architectures better suited for low-resource generalization, and extend the evaluation to interactive and communicative benchmarks that more directly reflect the ecological learning conditions motivating the BabyLM challenge.

Limitations

Our instruction tuning experiments are constrained by the relatively small size of the instruction tuning datasets compared to pre-training corpus, which may have reduced the impact of instruction-specific learning. A different allocation (using more instruction tuning data and proportionally less pre-training data) might yield stronger effects. Moreover, the balance between question–answering and conversational data is imperfect, with the latter underrepresented, potentially biasing results toward factual over interactive skills. Finally, the Simple Wikipedia augmentation process was only partially validated, and higher-quality or more diverse instructional sources could improve both robustness and generalization.

Acknowledgments

We acknowledge financial support under the PRIN 2022 Project Title "Computational and linguistic benchmarks for the study of verb argument structure" – CUP I53D23004050006 - Grant Assignment Decree No. 1016 adopted on 07/07/2023 by the Italian Ministry of University and Research (MUR). This work was also supported under the PNRR—M4C2—Investimento 1.3, Partenariato Esteso PE00000013—"FAIR—Future Artificial Intelligence Research"—Spoke 1 "Human-centered AI," funded by the European Commission under the NextGeneration EU programme"

References

Ruth A Berman and Dan Isaac Slobin. 2013. *Relating events in narrative: A crosslinguistic developmental study*. Psychology Press.

Noam Chomsky. 1980. Rules and representations. *Behavioral and brain sciences*, 3(1):1–15.

Andrea De Varda and Marco Marelli. 2023. Scaling in cognitive modelling: A multilingual approach to human reading times. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 139–149.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *CoRR*.

Selma Dündar-Coecke, Andrew Tolmie, and Anne Schlottmann. 2020. Children's reasoning about continuous causal processes: The role of verbal and non-verbal ability. *British Journal of Educational Psychology*, 90(2):364–381.

Jerry Fodor. 1975. The language of thought (new york: Thomas crowell).(1987a). *Psychosemantics: the Problem of Meaning in the Philosophy of Mind*.

Michael C Frank. 2023. Bridging the data gap between children and large language models. *Trends in Cognitive Sciences*, 27(11):990–992.

Alison Gopnik. 2011. The theory theory 2.0: probabilistic models and cognitive development. *Child Development Perspectives*, 5(3):161–163.

Valentin Hofmann, Leonie Weissweiler, David R Mortensen, Hinrich Schütze, and Janet B Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas Hikaru Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, and 1 others. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *CoRR*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Teresa McCormack and Christoph Hoerl. 2005. Children's reasoning about the causal significance of the temporal order of events. *Developmental Psychology*, 41(1):54.

Martina Miliani, Serena Auriemma, Alessandro Bondielli, Emmanuele Chersoni, Lucia Passaro, Irene Sucameli, and Alessandro Lenci. 2025. Explica:

Evaluating explicit causal reasoning in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

Carole Peterson and Allyssa McCabe. 1987. The connective 'and': Do older children use it less as they learn other connectives? *Journal of Child Language*, 14(2):375–381.

Jean Piaget. 2002. *Judgement and reasoning in the child*. Routledge.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? *Advances in neural information processing systems*, 36:55565–55581.

Paul Smolensky. 1988. On the proper treatment of connectionism. *Behavioral and brain sciences*, 11(1):1–23.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.

Michael Tomasello. 2009. *The cultural origins of human cognition*. Harvard university press.

Lev Semenovich Vygotsky. 1987. *The collected works of LS Vygotsky: Volume 1: Problems of general psychology, including the volume Thinking and Speech*, volume 1. Springer Science & Business Media.

Samuel R Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english (electronic resources).

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Zheyu Zhang, Han Yang, Bolei Ma, David Rügamer, and Ercong Nie. 2023. Baby's cothought: Leveraging large language models for enhanced reasoning in compact models. *arXiv e-prints*, pages arXiv–2308.

Xingmeng Zhao, Tongnian Wang, Sheri Osborn, and Anthony Rios. 2023. Babystories: Can reinforcement learning teach baby language models to write better stories? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 186–197.

Mask and You Shall Receive: Optimizing Masked Language Modeling For Pretraining BabyLMs

Lukas Edman[1,2] **Alexander Fraser**[1,2,3]

[1]School of Computation, Information and Technology, TU Munich
[2]Munich Center for Machine Learning
[3]Munich Data Science Institute

lukas.edman@tum.de

Abstract

We describe our strategy for the 2025 edition of the BabyLM Challenge. Our main contribution is that of an improved form of Masked Language Modeling (MLM), which adapts the probabilities of the tokens masked according to the model's ability to predict them. The results show a substantial increase in performance on (Super)GLUE tasks over the standard MLM. We also incorporate sub-token embeddings, finding that this increases the model's morphological generalization capabilities. Our submission beats the baseline in the `strict-small` track.

1 Introduction

Traditionally, language models (LMs) have required billions to trillions of tokens for training, much less than a human typically sees, all while still suffering from the inability to accomplish relatively trivial tasks for humans. The 3rd BabyLM Challenge (Charpentier et al., 2025), asks if we can train models more efficiently, making it more akin to the efficiency of human learning.

A notable difference in standard LM training from human learning is that schooling is typically organized in curricula, meanwhile LMs tend to train on data in an unstructured, random manner. Thus, it is natural to consider bringing the concept of curriculum learning to LM training. Several works have attempted this, with only minimal success being shown in BabyLM's tracks (Warstadt et al., 2023; Hu et al., 2024).

Our approach for this year's BabyLM returns to this ever-elusive goal of effectively incorporating curriculum learning by optimizing the Masked Language Modeling (MLM) objective used to train encoder models. MLM by default masks every token with equal probability, but this is likely not optimal. Certain tokens that are easy to predict are likely a waste of time to mask, while other tokens that are more difficult to mask may require the model to learn key language concepts in order to reliably predict them.

We introduce a form of MLM that adapts over the course of training, weighting the probabilities of masking individual tokens differently, based on the model's performance predicting them.

We also introduce an entirely different concept, designed instead to incorporate sub-token level information into the model's embeddings. Many works have shown the potential benefits of a model having access to sub-tokens or individual characters. While the evaluation tasks from previous BabyLM years did not require such finer-grained information to complete them, adjective nominalization was added as a task this year, alongside a similar task of converting to past tense, which was added as a hidden task for the final evaluation. We expect finer-grained character information to be useful for this task, especially.

2 Related Work

We focus on works related to our novel methods: Adaptive MLM and N-hot encodings.

2.1 Masked Language Modeling

A number of works have looked at improving the Masked Language Modeling objective. Wettig et al. (2023) experimented with the probability of masking a token, finding that values higher than the standard 15% worked well in certain tests. Yang et al. (2023) continued along these lines and found that higher probabilities work better early in training, and lower values are better later. They also adjusted the probabilities of words being masked based on their POS tag, arguing that some word classes are much easier to predict and thus a waste of training time. Belfathi et al. (2024) similarly weigh words based on their domain specificity in order to do domain adaptation.

The most similar work to ours is Zhang et al.

Proceedings of the First BabyLM Workshop, pages 457–465

(2023b), whose dynamic masking strategy works similarly to our soft approach. They also weigh tokens based on their respective loss, but instead with the explicit purpose of oversampling rare tokens. This is used as a further pre-training strategy for BERT, and shows some limited improvement.

In terms of cognitive plausibility, the adaptive method we introduce has some similarities to human behavior. In eye-tracking studies, humans tend to fixate on words that are more difficult to predict (Ehrlich and Rayner, 1981; Rayner and Well, 1996). EEG studies have similarly shown that unpredictable words require more cognitive effort to process (Kutas and Hillyard, 1984). Our method similarly steers the model to focus more on words that are difficult to predict.

2.2 Character-level Information

A number of works have sought to include character information in models. CharacterBERT (El Boukkouri et al., 2020), ByT5 (Xue et al., 2022), Byte Latent Transformer (Pagnoni et al., 2025), among many others have attempted to incorporate character or byte-level information within large-scale models. These works have noted that character-level models tend to train more efficiently, showing the normalized loss (bits-per-byte) can reach the same level in fewer steps, but they have not been extensively studied in a limited-resource setting such as BabyLM.

For BabyLM itself, Edman and Bylinina (2023) have attempted to first pretrain on a character-level vocabulary and swapping to a BPE vocabulary without much success. Goriely et al. (2024) trained phoneme-level models but did not find improvements on the BabyLM benchmarks. The lack of improvements could be due to BabyLM not sufficiently measuring the models' understanding of orthography, phonology, or other aspects that require finer-grained information within the inputs. New to this year however is the adjective nominalization and past tense tasks (Hofmann et al., 2025), which measure a model's morphological intuition by choosing the perceived correct ending to an imaginary adjective in order to convert it to a noun, or the perceived correct past-tense form of an imaginary infinitive verb.

3 Method

We first describe our adaptive masked language modeling (AMLM) scheme, then our token-level n-hot embedding architecture, and finally note the experimental details.

3.1 Adaptive MLM

The goal of AMLM is to improve the masking strategy such that we train the model on tokens from which it can learn the most. Therefore, tokens that are easy to predict should be assigned a lower probability of being masked. We employ 2 metrics to weigh each token: accuracy (**hard**) and loss (**soft**).

For both metrics, we start with a uniform probability for each token in the vocabulary:

$$w_{t=0,i} = p_{\text{mlm}}, \quad \forall i \in V \tag{1}$$

where p_{mlm} is the overall probability of a token being masked in any given sequence, typically 15%. For every batch, we record the statistics of whether the model correctly predicted the masked tokens and the token-level loss of masked tokens. At the start of every timestep t (which we define as 200 batches), we update the probabilities:

$$w_{t,i} = \lambda\, w_{t-1,i} + (1-\lambda)\, \tilde{w}_{t,i} \tag{2}$$

$$\tilde{w}_{t,i} = p_{\text{mlm}}\,(1 - \text{score}_{\text{t-1,i}}) \tag{3}$$

where $\text{score}_{t,i}$ is a scoring function, using either the hard or soft metric. We set $\lambda = 0.2$ empirically, which weighs the most recent statistics highly, but accumulates with previous timesteps nonetheless. For the accuracy-based **hard** metric, our scoring function is a smoothed accuracy:

$$\text{score}_{t,i} = \frac{\text{correct}_{t,i} + 0.5}{\text{total}_{t,i} + 1} \tag{4}$$

where $\text{correct}_{t,i}$ and $\text{total}_{t,i}$ refer to the number of correctly predicted tokens of type i at timestep t, and the total number of predicted tokens. For the loss-based **soft** metric, the scoring function is a normalized, inverted loss:

$$\text{score}_{t,i} = 1 - \text{norm}(\ell_{t,i}) \tag{5}$$

with $\ell_{t,i}$ corresponding to the average cross-entropy loss of token i for timestep t.

We allow the scores to range between 0 and 1, so that if the model is perfect at predicting a token (a score of 1), the probability of masking said token tends to 0. Meanwhile, a score of 0 causes the probability to tend towards p_{mlm}. Finally, when masking each input sequence, the probabilities per token are normalized such that the average is p_{mlm}, allowing individual tokens' mask probabilities to exceed p_{mlm}.

3.2 Token-level N-hot Embeddings

Another strategy we try is incorporating more character-level information into the input embeddings. We accomplish this by what we call token-level n-hot embeddings, and it is best illustrated with an example: For the token *_doing*, we get all of the substrings that are also in our vocabulary, e.g., *_doin*, *g*, *_do*, *ing*, etc. These substrings are then encoded as an n-hot feature vector, i.e., 1 for *_doin*, *g*, *_do*, *ing*, etc., and 0 for everything else (hence the name n-hot). We then project this encoding into the embedding space with a linear layer, and add that to a separate, standard token embedding. To make this efficient, all of the n-hot encodings can be pre-calculated, with only the linear layer being trained on-the-fly, making the increase in training time negligible.

This strategy should be useful for any tasks that involve sub-token information. In particular, the adjective nominalization task asks for the model to provide the most plausible nominalization to a made-up adjective, e.g., "wugable" → "wugability". The ability for token-level n-hot encodings to trivially encode morphemes should make this task easier.

3.3 Experimental Setup

Our setup most closely follows that used by the strict-small GPT-BERT baseline[1], using the same learning rate, optimizer, and batch size. We use the same hidden and intermediate size for our model, however we use the Deberta-V2 (He et al., 2021) architecture instead, given its ease of use and overall similarity to GPT-BERT, having the same attention mechanism. We use a starting sequence length of 64 and raise it to 256 after 5 epochs. We use BPE with byte-fallback and a vocabulary size of 40k, following Edman and Bylinina (2023)'s findings that 40k appears near optimal. In terms of the probability of masking a token, we experiment with a decaying mask as suggested by Yang et al. (2023), opting for 40% at the start and linearly decaying to 15%. We compare this to the standard constant 15%. All of the hyperparameters are listed in Appendix A.

In terms of data, we use the same data as in Edman et al. (2024), which consists of the initial BabyLM data, with the child-directed speech removed, and replaced with data from Zhang et al. (2023a). Their data is synthetically generated triplets of sentences, paraphrases, and contradictions. We only use the data, not their contrastive learning approach. We compare this dataset to the original dataset from the shared task, as well as vocabulary size, in Appendix B.

[1] https://huggingface.co/BabyLM-community/babylm-baseline-10m-gpt-bert-masked-focus

4 Results

We first present our results using our MLM objective. In Table 1, we see that, overall, AMLM with the hard metric and decaying mask performs best overall. In general, the hard method performs better than the soft as well as regular MLM. While the differences are not very large, the hard mask strategy performs consistently better across multiple runs.

In terms of using a decaying mask versus a constant one, the results are not as immediately clear, mainly due to adjective nominalization having a strong effect on zero-shot performance. In head-to-head comparisons for zero-shot, the decaying mask is better in 55% (15/27) of tasks, tied in 7% (2/27), and worse in 37% (10/27). Combined with the better performance in fine-tuning, this confirms evidence from Yang et al. (2023).

4.1 N-hot Encodings

We show the scores for the model with n-hot encodings also in Table 1. We show only results with n-hot alongside the hard method, as it is the most performant, but the n-hot encodings are compatible with any form of pretraining.

On average, the n-hot encodings do not appear better or worse. Focusing on specific tasks, they perform noticeably worse on BLiMP. We suspect this is due to a sub-optimal manner of combining the n-hot embeddings with the regular ones. The performance on the (Super)GLUE tasks is comparable, suggesting that fine-tuning this model for a task may still yield competitive results. The performance on adjective nominalization is the most promising. There, we see the performance increase from 10 to 30 points over the comparable hard methods.

It is not entirely clear why we do not see the same increase with the past-tense task. This may be due to the past-tense examples including more options, lowering the chance of human agreement, the options being more irregular, e.g. the past tense of "veed" could be "veeded", "ved", or "vode", or there being more collisions with real words, e.g.

		Constant Mask			Decaying Mask			Hard + N-Hot	
		Reg	Hard	Soft	Reg	Hard	Soft	Const	Decay
Zero-shot	BLiMP	70.7	70.0	69.7	70.8	**71.3**	71.4	68.1	67.7
	Supplement	55.5	56.9	56.2	57.8	**58.3**	58.1	56.2	56.4
	EWoK	50.6	49.9	50.7	50.2	**50.9**	50.1	50.5	50.2
	Eye-tracking	9.0	9.2	**9.4**	9.1	8.9	8.8	8.6	8.5
	Self-paced Reading	**4.2**	4.0	4.0	4.1	3.7	4.0	3.9	3.8
	Entity Tracking	42.9	43.8	42.9	**44.5**	44.4	39.3	43.3	34.5
	Adj Nominalization	35.3	34.3	22.0	11.7	14.3	0.0	**43.7**	42.0
	Past-tense	4.0	**6.3**	-6.7	1.3	1.3	1.7	-0.3	4.0
	COMPS	52.6	53.1	53.2	53.9	**54.0**	53.2	52.0	52.4
	Zero-shot Avg	36.1	**36.4**	33.5	33.7	34.1	32.2	36.2	35.5
Finetune	BoolQ	69.7	68.6	69.7	**70.2**	69.2	69.6	67.3	**70.2**
	MNLI	59.1	59.9	59.3	61.8	62.3	61.6	59.3	**62.5**
	MRPC	88.1	89.6	88.2	89.8	**90.6**	89.8	85.7	89.1
	QQP	72.5	72.6	72.4	72.9	**73.1**	72.9	72.2	72.8
	MultiRC	68.1	65.2	**68.3**	64.6	68.2	67.9	64.4	68.4
	RTE	57.8	60.2	58.0	62.1	**64.3**	62.6	62.1	60.2
	WSC	64.1	66.7	64.1	66.7	63.5	**67.3**	**67.3**	66.0
	Finetune Avg	68.5	70.0	68.6	69.4	**70.1**	69.9	68.3	69.9

Table 1: Results of Regular MLM versus Hard and Soft AMLM, averaged over 3 runs. We also show performance when adding n-hot encodings using hard AMLM.

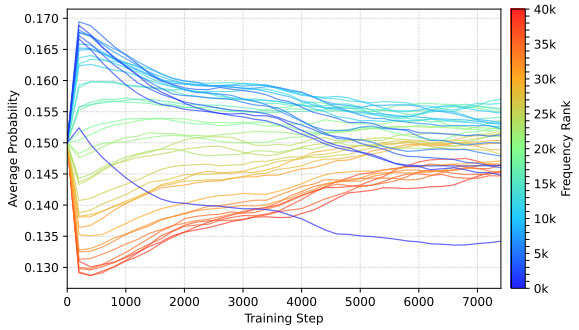


Figure 1: Token masking probabilities for hard method, grouped by frequency rank (in groups of 1000). Lower rank indicates higher frequency (e.g., blue is the most frequent group of words).

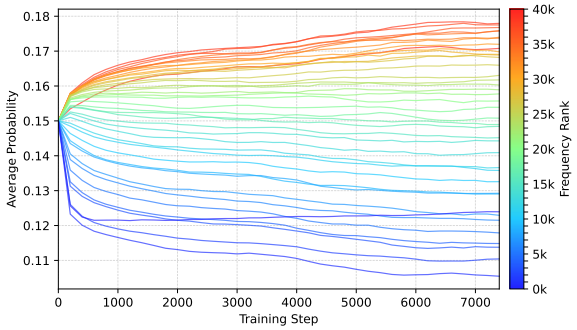


Figure 2: Token masking probabilities for soft method, grouped by frequency rank (in groups of 1000). Lower rank indicates higher frequency.

"scor" possibly becoming "scored". Ultimately, these tasks have no objectively correct answer. It would be better to measure the performance of n-hot encodings and similar methods that incorporate character-level information on tasks that require such information, like morphological inflection, for example.

4.2 AMLM Analysis

As we record the statistics of the model's masked-token predictions, we can analyze the fluctuations of the probabilities with respect to various properties of words. We focus on two here: frequency and part-of-speech (POS).

In Figure 1, we can see the average probabilities of masking words using the hard method, grouped by frequency, in bins of 1000. Here, we see that initially, more frequent tokens are weighted higher, and rarer tokens lower. The weights of the top 5000 tokens quickly drop down, but the middle 20000 end up generally higher. The bottom 10000 tokens rise in weight but not back to 15%.

Figure 2 shows the soft masking probabilities, with a similar trend over time but quite different initial steps. The start sees common tokens quickly dropping in probability and rare words rising. The stark difference is due to the continuous nature of the soft method versus the discrete nature of the hard method. In the soft method, the loss goes down steadily, which immediately affects the probabilities. In the hard method, the accuracy does not

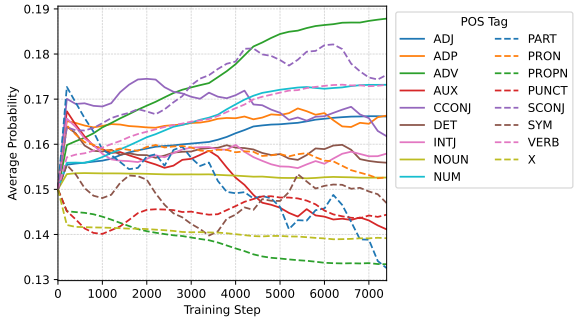


Figure 3: Masking probabilities by POS tag for hard method.

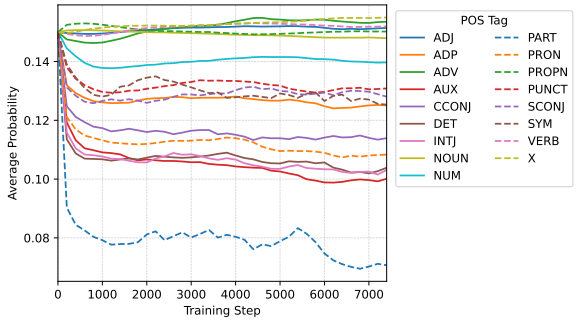


Figure 4: Masking probabilities by POS tag for soft method.

immediately go up, as it takes time for the model to promote the correct token to the highest probability. The smoothing applied in Equation 4 causes the more common words to be favored for masking, hence the steep increase in the beginning.

Given that the hard method performs slightly better overall, these results indicate that masking common words more often at the start, even more often than the standard MLM does already due to their frequency in the text, may be a useful strategy for training more efficiently with MLM. As noted in Section 2, our soft approach is similar to the strategy used by Zhang et al. (2023b), whose goal was to mask rarer tokens. Although the settings differ, our results may explain the limited improvement seen in their work.

In Figure 3, we can see the hard masking probabilities grouped by POS tag. We can see that the probabilities per POS tag are not uniform; instead, adverbs, subordinating conjunctions, numbers, and verbs are given higher weights. This tracks in general with intuition: adverbs have a lot of flexibility in their usage, as they can modify verbs, adjectives, or other adverbs, making them difficult to predict. Subordinating conjunctions being weighted higher might indicate that the model has difficulty understanding logical connections between multiple clauses. Exact numbers can also be difficult to predict.

Meanwhile, we see that particles, proper nouns, and X are weighted lower. The most common particle, "to", should be easy to predict when the model can see that the next token is an infinitive verb. Proper nouns may also be relatively easy, as their usage is usually very context-dependent. X comprises mainly of subtokens, e.g., "arrog", which would typically be part of "arrogant" or "arrogance", but is split by BPE due to the limited vocabulary size.

In Figure 4, we see a very different story: X and proper nouns are weighted highly while most other classes are weighted lower. It may seem as if the average does not equal 15%, but X and PROPN, along with NOUN, are the 3 most common token classes in the vocabulary, making up 8.4k, 9.2k, and 12.4k, respectively.

The fact that we are POS tagging on the token level brings the unfortunate side effect of grouping several word parts into the X category. However, given that the X category, along with proper nouns, constitutes the majority of the difference, it is interesting to speculate why. Given their high weight for the soft method and low weight in the hard method, these words are often correctly predicted, but with a high loss. This is likely due to there being several candidate tokens that would fit the sentence. For example, "arrog" could easily be replaced with 'ignor" without much change in meaning. Further analysis is needed for a better understanding of this difference.

4.3 Submission

Here we compare our submitted models to the leaderboard with the strongest baseline provided by the organizers. We select only models trained with RNG seed 0 so as not to overfit to the evaluation benchmark. The results are shown in Table 2.

Here, we can see that our hard decay model and hard n-hot models outperform the baseline according to the aggregate score. For the hard decay model, the main improvement appears to be the (Super)GLUE scores, however these are aggregated into one value for the final score, so their weighting is less important. Entity tracking seems to be the only other culprit, as the rest of the scores are fairly close to each other.

The n-hot model stands out in its performance on adjective nominalization, which is enough for

		BertGPT	AMLM		
		Masked MNTP	Hard	Hard Decay	N-hot Hard
Zero-shot	BLiMP	70.4	69.9	**71.4**	65.6
	Supplement	**63.7**	57.9	59.2	54.8
	EWoK	50.0	50.0	**51.0**	49.7
	Eye-tracking	**9.4**	8.9	8.3	8.1
	Self-paced Reading	3.4	**4.0**	3.5	3.3
	Entity Tracking	40.0	43.6	**44.2**	43.7
	Adj Nominalization	2.7	41.0	34.0	**64.0**
	Past-tense	**28.7**	8.0	6.0	1.0
	COMPS	53.5	52.3	**54.2**	51.3
	AoA	0.3	-15.0	-0.9	**16.3**
	Zero-shot Avg	32.2	32.1	33.1	**35.9**
Finetune	BoolQ	67.6	69.2	**69.5**	68.9
	MNLI	51.4	59.3	**62.3**	58.5
	MRPC	86.1	**90.3**	**90.5**	81.0
	QQP	67.4	72.4	**73.1**	72.2
	MultiRC	**71.6**	69.8	68.6	57.5
	RTE	57.5	61.1	**63.3**	64.0
	WSC	61.5	**73.1**	63.5	69.2
	Finetune Avg	66.2	**70.7**	70.1	67.3
	Final Score	38.2	34.2	38.3	**41.9**

Table 2: Our submission models (AMLM) versus the comparable BabyLM baseline (BertGPT). Final score refers to the scoring equation used by the BabyLM evaluation leaderboard.

the average on zero-shot tasks to favor this model, as well as the aggregate score. As the n-hot model performs poorly on most other tasks, the final scoring metric seems to be unfairly skewed. We further question the efficacy of the BabyLM metrics in Appendix C, where we find that a model whose training loss spikes due to a mistake on our part actually gets a higher aggregate score than any model introduced so far.

5 Conclusion

The BabyLM Challenge challenges us to train language models on a limited data budget and, for the first time this year, a limited training time budget (by way of epochs). We show that greater training efficiency can be achieved through Adaptive MLM, which changes the probabilities of tokens being masked during training, according to their difficulty. The results show an increase in performance, beating the baseline set by the organizers. We also investigate a method of incorporating subtoken-level information into the model, which showed promising performance on adjective nominalization, the task that requires finer-grained, morpheme-level understanding.

There is plenty of future work to be investigated. The adaptive masking scheme required setting several hyperparameters empirically, and is likely far from optimized. While we use generic statistics based on individual tokens, groups of tokens being masked in tandem may be more challenging and force the model to learn more properties of language. For this, we would suggest word-level or phrase-level masking, or more interestingly, a neural masker, which learns to mask tokens based on what is most challenging for the main model to predict.

Our subtoken approach is also far from optimal. In theory, it should be possible to incorporate two or more granularities of input without any negative effects on downstream performance. Such improvements would have a considerable impact not only in the BabyLM sphere, but higher-resource NLP as well.

Acknowledgements

We thank BabyLM anonymous reviewers for the helpful comments. The work was supported by the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No. 101113091) and by the German Research Foundation (DFG; grant FR 2829/7-1).

References

Anas Belfathi, Ygor Gallina, Nicolas Hernandez, Richard Dufour, and Laura Monceaux. 2024. Language model adaptation to specialized domains through selective masking based on genre and topical characteristics. *Preprint*, arXiv:2402.12036.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Lukas Edman and Lisa Bylinina. 2023. Too much information: Keeping training simple for BabyLMs. pages 89–97, Singapore.

Lukas Edman, Lisa Bylinina, Faeze Ghorbanpour, and Alexander Fraser. 2024. Are BabyLMs second language learners? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 166–173, Miami, FL, USA. Association for Computational Linguistics.

Susan F Ehrlich and Keith Rayner. 1981. Contextual effects on word perception and eye movements during reading. *Journal of verbal learning and verbal behavior*, 20(6):641–655.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. pages 6903–6915, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Marta Kutas and Steven A Hillyard. 1984. Brain potentials during reading reflect word expectancy and semantic association. *Nature*, 307(5947):161–163.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. Byte latent transformer: Patches scale better than tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.

Keith Rayner and Arnold D Well. 1996. Effects of contextual constraint on eye movements in reading: A further examination. *Psychonomic Bulletin & Review*, 3(4):504–509.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. pages 1–34, Singapore.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? pages 2985–3000, Dubrovnik, Croatia.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Dongjie Yang, Zhuosheng Zhang, and Hai Zhao. 2023. Learning better masking for better language model pre-training. pages 7255–7267, Toronto, Canada.

Junlei Zhang, Zhenzhong Lan, and Junxian He. 2023a. Contrastive learning of sentence embeddings from scratch. pages 3916–3932, Singapore.

Linhan Zhang, Qian Chen, Wen Wang, Chong Deng, Xin Cao, Kongzhang Hao, Yuxin Jiang, and Wei Wang. 2023b. Weighted sampling for masked language modeling. *Preprint*, arXiv:2302.14225.

A Training Hyperparameters

We note the complete hyperparameters used in Table 4.

	BertGPT Baselines		AMLM - Hard Decay		
	Causal	Masked	Success	Failure	Untrained
BLiMP	**71.7**	70.4	71.4	59.0	48.8
BLiMP Supplement	63.2	**63.7**	59.2	51.7	42.3
EWoK	49.5	50.0	51.0	**56.0**	50.0
Entity Tracking	34.6	40.1	**44.2**	41.3	41.7
Adj Nominalization	59.2	2.7	22.3	**78.1**	77.4
Past Tense	12.9	**28.7**	6.2	-12.1	-16.2
COMPS	52.8	53.6	54.2	**82.2**	50.0
Reading	**6.7**	6.4	5.9	6.4	5.7
AoA	-3.9	0.3	-0.9	**34.2**	0.0
(Super)GLUE	65.1	66.0	**69.8**	57.7	62.1
Avg	41.2	38.2	38.3	**45.4**	36.2

Table 3: Baselines and our best submitted model, compared to a failed run and untrained model.

	Parameter	Value
Model	Architecture	Deberta
	Hidden Size	384
	Intermediate Size	1280
	Dropout	0.1
	Vocabulary Size	40000
Training	Sequence Length	64, 256
	Batch Size (in tokens)	16384
	Learning Rate	7e-3
	Epochs	10
	Number of Steps	8325
	Scheduler	Cosine
	Warmup Ratio	1%
	Mask Ratio	0.4→0.15
	Random Ratio	0.1
	Keep Ratio	0.1
	Weight Decay	0.01
	Optimizer	LAMB
	Optimizer Epsilon	1e-8
	Optimizer Beta 1	0.9
	Optimizer Beta 2	0.95
	Grdient Clipping	1

Table 4: Hyperparameters used (except where otherwise indicated).

B Ablation of Dataset and Vocabulary Size

Two of the major differences between our models and the baselines trained by the organizers are the training dataset and vocabulary size. We ablate the two with a standard MLM training scheme and our hard AMLM method in Table 5.

First, concerning the dataset, the new dataset appears generally better for the finetuning tasks. Most of this increase is in the MNLI, MRPC, and QQP tasks. This aligns with the findings of Edman and Bylinina (2023), and can be explained by the additional data having similar content (i.e., paraphrases).

As for vocabulary size, the results appear dependent on the dataset. For the new dataset, the higher vocabulary size appears to perform better, especially for BLiMP and Entity Tracking. The higher vocabulary size may be slightly more optimal for the new dataset, but for the original dataset, the lower vocabulary size appears more optimal. The difference in zero-shot performance on adjective nominalization and past-tense is quite substantial, though the variance in these metrics is high, so it is difficult to gauge the importance of these results.

C Should we trust BabyLM Metrics?

During our experimentation, we encountered some surprising results with respect to some of the metrics used to evaluate models. First, we found that one could obtain quite high scores on adjective nominalization, around 78, just from initialization, no training required.

Furthermore, we accidentally trained a model with too low a batch size for the corresponding learning rate, causing the loss to spike during

		Regular				AMLM - Hard			
		Original		New		Original		New	
		16k	40k	16k	40k	16k	40k	16k	40k
Zero-shot	BLiMP	66.9	66.4	67.2	70.7	67.5	67.0	68.8	70.0
	Supplement	58.9	59.2	58.3	55.5	62.0	59.0	57.9	56.9
	EWoK	49.9	50.6	49.0	50.6	51.2	50.8	50.3	49.9
	Eye-tracking	6.2	6.3	7.2	9.0	6.5	5.9	6.7	9.2
	Self-paced Reading	3.6	3.2	4.0	4.2	3.5	4.0	3.6	4.0
	Entity Tracking	31.2	27.5	32.6	42.9	32.1	35.6	36.9	43.8
	Adj Nominalization	37.0	18.0	56.0	35.3	50.0	11.0	43.0	34.3
	Past-tense	29.0	18.0	0.0	4.0	23.0	-10.0	18.0	6.3
	COMPS	51.4	52.0	51.9	52.6	51.9	51.9	52.7	53.1
	Zero-shot Avg	37.1	33.5	36.2	36.1	38.6	30.6	37.5	36.4
Finetune	BoolQ	68.5	67.8	70.9	69.7	70.2	68.4	68.8	68.6
	MNLI	44.0	47.3	56.6	59.1	44.5	48.5	57.9	59.9
	MRPC	82.6	82.0	86.7	88.1	82.7	83.7	88.7	89.6
	QQP	66.3	67.8	71.6	72.5	67.0	68.8	72.7	72.6
	MultiRC	67.1	66.6	66.6	68.1	66.2	64.4	67.8	65.2
	RTE	59.7	59.0	59.0	57.8	56.8	53.2	61.1	60.2
	WSC	65.4	65.4	61.5	64.1	63.5	61.5	65.4	66.7
	Finetune Avg	64.8	65.1	67.5	68.5	64.4	64.1	68.9	69.0

Table 5: Ablation of vocabulary size and dataset.

training and never recover. While it performs expectedly poorly on some metrics, such as BLiMP and (Super)GLUE, it performs remarkably well on others, namely EWoK, adjective nominalization (again), COMPS, and AoA. We show the results in Table 3.

This raises the question: should we trust the metrics used in BabyLM? For BLiMP and (Super)GLUE the answer appears to be yes. Failed and untrained models perform expectedly poorly on these. For self-paced reading and eye-tracking, the numbers appear to stay relatively similar, regardless of the model. And for the rest, their scores should probably be taken with a baby fist of salt.

Once Upon a Time: Interactive Learning for Storytelling with Small Language Models

Jonas Mayer Martins **Ali Hamza Bashir**
Muhammad Rehan Khalid **Lisa Beinborn**

University of Göttingen, Institute of Computer Science, Germany
firstname.lastname@uni-goettingen.de

Abstract

Children efficiently acquire language not just by listening, but by interacting with others in their social environment. Conversely, large language models are typically trained with next-word prediction on massive amounts of text. Motivated by this contrast, we investigate whether language models can be trained with less data by learning not only from next-word prediction but also from high-level, cognitively inspired feedback. We train a student model to generate stories, which a teacher model rates on readability, narrative coherence, and creativity. By varying the amount of pretraining before the feedback loop, we assess the impact of this interactive learning on formal and functional linguistic competence. We find that the high-level feedback is highly data efficient: With just 1 M words of input in interactive learning, storytelling skills can improve as much as with 410 M words of next-word prediction.

Models and data | Code repository

1 Introduction

HUMANS are storytelling animals (Gottschall, 2012; Campbell, 2008). From early myths to modern science, narratives have served not only as entertainment but also as cognitive tools to make sense of the world. Scientific models and historical accounts, personal and collective identities, and even abstract institutions such as currency, law, and national borders can all be understood as shared stories (Bruner, 1991). Through our capacity for language, we establish a communicative common ground to align intentions, construct shared realities, and thus cooperate at societal scales (Tomasello, 2008, 2014; Clark and Schaefer, 1989; Clark and Brennan, 1991).

In recent years, language models have achieved surprising proficiency in generating natural language. However, training these artificial neural networks with billions to trillions of parameters is inefficient (Wilcox et al., 2025). While modern supercomputers are trained on the order of 10^{13} words (DeepSeek-AI, 2025), a child is exposed to between 10^8 and 10^9 words by age 13, extrapolating from Gilkerson et al. (2017). How do children acquire language so efficiently? In this work, we explore one potential ingredient: enriching the learning signal for language models beyond classical next-word prediction (Stöpler et al., 2025).

Artificial and biological neural networks differ in structure and dynamics, yet both can acquire complex linguistic behavior (Evanson et al., 2023). The standard training objective for language models—next-word prediction—superficially resembles predictive processing (Clark, 2013; Ryskin and Nieuwland, 2023), but does not reflect the rich, interactive learning experienced by children. We hypothesize that incorporating high-level feedback can guide language models toward more efficient functional linguistic competence, i.e., coherent, pragmatic, and creative use of language (Mahowald et al., 2024).

While the human brain excels at finding patterns in sensory input—a capacity central to early language learning (Saffran, 2020)—children are more than just passive recipients of this input. Instead, they learn language in a social context, shaped by interaction and feedback from caregivers (Tomasello, 2008; Clark, 2018). This feedback includes both implicit cues, such as contingent responses and repetitions, and explicit forms, such as corrections and confirmations (Cheatham et al., 2015; Nikolaus and Fourtassi, 2023).

By contrast, traditional language modeling is fully self-supervised. External feedback is integrated only later, during fine-tuning for applied tasks, when the model receives feedback from labeled examples (Parthasarathy et al., 2024). More recently, reinforcement learning (RL) has been introduced to language modeling to better align

Proceedings of the First BabyLM Workshop, pages 466–480

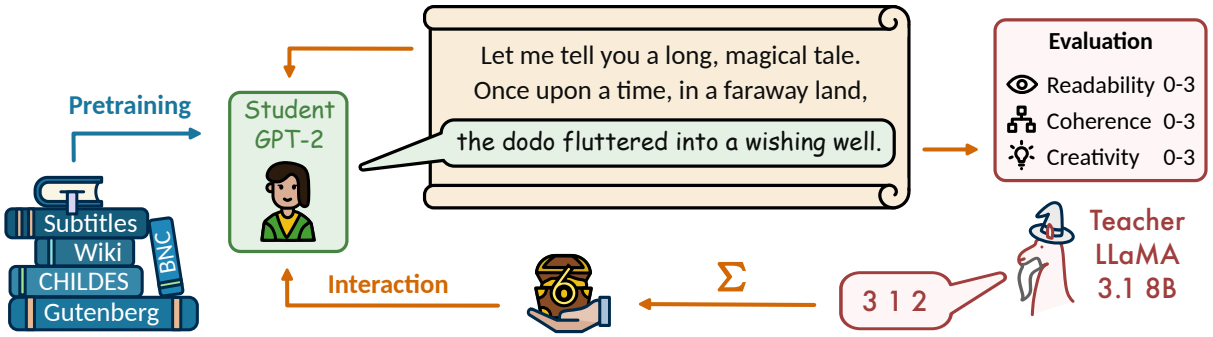


Figure 1: Schematic of the interactive learning setup with storytelling feedback. During pretraining, the student model optimizes next-word prediction on the BabyLM corpus. In the interaction stage, the student model completes a story prompt. A teacher model then evaluates the story on three criteria using a Likert scale from 0 to 3. The student receives the sum of these scores as a reward and updates its parameters to generate stories that maximize the expected reward.

model outputs with human preferences.

In this work, we replace part of the next-word prediction in pretraining by reinforcement learning in interaction with a teacher model, employing storytelling as a task that requires functional linguistic competence, see Fig. 1. After pretraining on the BabyLM corpus (Charpentier et al., 2025), the student model enters the interaction loop: First, the student generates a story from a generic snippet. Next, the teacher model judges the generated story with respect to readability, narrative coherence, and creativity. Finally, the student model receives the sum of the teacher scores as a reward and updates its parameters to maximize the expected reward.

We assess how high-level narrative and linguistic feedback impacts the student model's learning dynamics. Specifically, we demonstrate that partially replacing next-word prediction with interaction augments storytelling ability without compromising low-level linguistic generalization. Remarkably, with less than 1 M input words of interactive learning, storytelling skills improve as much as 410 M additional words of conventional pretraining. Finally, we examine how the amount of pretraining influences the effectiveness and dynamics of reinforcement learning for storytelling.

2 Interactive learning for small language models

Prior work on data efficiency in language modeling motivates alternative training objectives. Discussing storytelling as a lens for evaluating linguistic competence, we present interactive learning as a cognitively inspired approach to improving data efficiency and functional language skills in small models.

2.1 Scaling and parsimony

Large language models generally perform better with more parameters and more training data (Bahri et al., 2024). From a cognitive perspective, data parsimony is of particular interest. A child encounters orders of magnitude fewer words than large language models: Extrapolating from Gilkerson et al. (2017), we estimate that by age 13 a child has been exposed to around 100 million to 1 billion words—only a fraction of the input given to modern language models. Inspired by how children acquire language,[1] the BabyLM Challenge seeks to close this gap in data efficiency (Warstadt et al., 2023; Hu et al., 2024; Charpentier et al., 2025). The findings from previous BabyLM challenges show that the most promising improvements in model performance come from changes in architecture and training objective (Warstadt et al., 2023; Hu et al., 2024).

We hypothesize that the next-word prediction objective—operating at the word or subword level—is too fine-grained to foster sufficient abstraction. In addition, next-word prediction requires multiple exposures to each word for effective learning and introduces frequency biases and anisotropy in model representations (Diehl Martinez et al., 2024; Godey et al., 2024). Achieving greater data efficiency may require a more comprehensive signal that incorporates high-level feedback.

2.2 Modeling storytelling

Humans communicate through stories and improve as storytellers by learning from interactive feedback. As a learning objective, storytelling is partic-

[1]We use language *learning* and *acquisition* interchangeably in this work.

ularly valuable because it requires *functional linguistic competence* (i.e., pragmatic use of language in real-world situations), as opposed to *formal linguistic competence* (i.e., knowledge of linguistic rules and patterns) (Mahowald et al., 2024). However, what defines a good story is difficult to formalize (Chhun et al., 2022) and existing metrics align poorly with human judgments (Guan et al., 2021).

Contemporary language models can produce fluent and grammatically correct stories but frequently struggle with coherence, creativity, and narrative structure (See et al., 2019; Xie et al., 2023). For example, models often fail at *entity tracking* (keeping track of facts about the world in a story), which is crucial for coherent stories (Kim and Schuster, 2023; Li et al., 2021). We propose that these functional skills can be improved by enriching the training objective with storytelling feedback.

2.3 Interactive learning

In multi-agent signaling games, the interactions of agents can lead to the emergence of novel communication protocols or even languages (Boldt and Mortensen, 2024; Bernard et al., 2024; Lazaridou et al., 2020). Also, cognitively inspired feedback can improve model performance (Nikolaus and Fourtassi, 2021; Saha et al., 2023; Stöpler et al., 2025). While previous work has explored various forms of feedback, our approach lets the student model generate stories freely in response to a writing prompt, while the teacher model provides high-level feedback on story quality.

Reinforcement learning, although a well-established method in machine learning, is relatively new to natural language processing (Parthasarathy et al., 2024; Havrilla et al., 2024). With regard to storytelling, reinforcement learning of sufficiently pretrained models appears surprisingly robust to sparse reward signals (Zhao et al., 2023; Wu et al., 2025). Unlike knowledge distillation, which approximates the function of a large language model through a model with fewer parameters (Dasgupta et al., 2023), our method uses textual feedback rather than probability distributions. This approach may be less computationally efficient, but it provides a more developmentally plausible reward signal, emulating student-teacher or child-caregiver interaction.

3 Methodology

As illustrated in Fig. 1, we model interaction as follows: A pretrained *student model* generates a story, which a *teacher model* then rates based on *evaluation instructions*. The teacher's scores serve as the reward signal for reinforcement learning via proximal policy optimization (PPO) (Parthasarathy et al., 2024).

Baselines We compare the student model against two baselines from the 2025 BabyLM challenge (Charpentier et al., 2025):

`1000M-pre` baseline: trained on 100 M unique words of the BabyLM corpus for 10 epochs with next-word prediction.

`SimPO` baseline: trained for 7 epochs with next-word prediction on the BabyLM corpus and 2 epochs interleaving next-word prediction with reinforcement learning. The reward is based on how similar the story completions of the student are to that of the teacher, providing corrective feedback.

Student model For our experiments, we use the same `GPT-2-small` architecture as the baseline for the student model and similar hyperparameters, see Appendix E.1. We divide the training into two stages:

`900M-pre` baseline: To stay within a word budget of 100 M words per epoch, we pretrain first on 90 % of the 100 M BabyLM corpus for 10 epochs.

`900M-RL` model: Subsequently, we do interactive learning with 1 M words of input. This yields fewer input words to the student model than the other baselines, namely, 901 M and 1,000 M words, respectively.

Teacher model Evaluating the quality of a story is a difficult task that requires both accurate judgments and computational efficiency. Based on pilot experiments, we select `Llama 3.1 8B Instruct` (Grattafiori, 2024).[2]

To mirror the student-teacher analogy, we keep the teacher model fixed throughout training.

Story generation To obtain a viable reward signal in reinforcement learning, we must elicit story-like outputs from the student model. We use the archetypal storytelling opening:

[2]Out of the three Llama Instruct models available for the Interaction Track of the BabyLM challenge (3.1 8B, 3.2 3B, and 3.2 1B), the largest one (Llama 3.1 8B Instruct) provides story scores with a reasonably high signal-to-noise ratio that aligned best with the developers' assessments of the story.

Student Model Input

Let me tell you a long, magical tale.
Once upon a time, in a faraway land,

Teacher feedback Defining the quality of a story is notoriously challenging (Chhun et al., 2022). Following Guan et al. (2021), we let the teacher model evaluate the student story on three criteria: readability, narrative coherence, and creativity.

Careful optimization of the teacher instructions was required for a strong and accurate learning signal, as language models are often highly sensitive to prompt phrasing (Chhun et al., 2022) and prone to label-induced biases (Saraf et al., 2025). During development, we refined the instructions to discourage shortcutting and ensure alignment with human judgment. We use rubrics to anchor the teacher's responses and provide examples of expected outputs. For each criterion, the teacher assigns a score from 0 (worst) to 3 (best), yielding robust and concise feedback. The full evaluation instructions are given in Appendix D.

Reward We use PPO to optimize the language model's policy for maximum expected reward. The reward R is calculated by combining the teacher scores $s_i \in \{0, 1, 2, 3\}$ for the three criteria i with a story length incentive based on the number of generated words L:

$$R = \frac{1}{1+\alpha}\left[\frac{1}{9}\sum_{i=1}^{3} s_i + \alpha \frac{L}{L_{\max}}\right] + r_{\mathrm{KL}}\,, \quad (1)$$

$L_{\max} = 100$ is the maximum allowed number of subword tokens (to normalize length), and $\alpha = 0.4$ controls the relative weight of the length bonus. The Kullback–Leibler (KL) divergence r_{KL} prevents the trained model from diverging too far from the pretrained baseline. See Appendix E.2 for full training parameters.

Experimental setup We first pretrain the `GPT-2-small` student model on 90 % of the BabyLM corpus for 10 epochs. To track the learning dynamics, we save checkpoints at logarithmically spaced intervals (1 M, 2 M, …, 10 M, 20 M, …, 100 M, 200 M, …, and 900 M words seen by the model). The final checkpoint constitutes our `900M-pre` baseline.

To assess the amount of pretraining necessary for efficient RL, we start the reinforcement learning from selected checkpoints (20 M, 50 M, 90 M, 200 M, 500 M, 900 M)[3] and train for 1 M words in 331.2k interactions (that is, 331.2k stories told), with evaluation checkpoints every 100k words.

During reinforcement learning, we log the stories, story length, teacher scores, as well as the KL divergence. The figures in Section 5.2 report Gaussian-smoothed batch averages ($\sigma = 30$ with batch size 360), unless otherwise noted.

4 Evaluation setup

We use the evaluation pipeline of the 2025 BabyLM Challenge (Charpentier et al., 2025). It comprises nine zero-shot diagnostic benchmarks and seven task-specific datasets that require model fine-tuning (see Appendix A).

Zero-shot diagnostics This suite evaluates the linguistic and conceptual capabilities of the language model by comparing its language modeling probabilities to human judgments. Minimal pair tasks are used to assess whether the model assigns higher probability to the more acceptable sentence. Each pair consists of two minimally contrastive sentences that isolate a certain phenomenon relating to syntactic and semantic grammaticality (`BLiMP`), dialogue and question processing (`BLimP supplement`), world knowledge about physical and social concepts (`EWoK`), and property inheritance (`COMPS`). In addition, the probabilities are correlated with human ratings for morphological properties of pseudo-words (`WUGs`), and to age-of-acquisition labels (`AoA`). Context integration capabilities of the model are tested by evaluating the proportion of the variance in eye-tracking (`Eye-T`) and self-paced reading (`SPR`) signals that is predictable from the surprisal of the model and by the accuracy of predicting the final state of an entity (entity tracking, `ET`) after a series of operations described as natural language discourse.

Task-specific fine-tuning The applicability of the model for downstream tasks is evaluated by its task-specific accuracy after supervised fine-tuning for question answering (`BoolQ` and `MultiRC`), natural language inference (`MNLI` and `RTE`), paraphrase recognition (`MRPC` and `QQP`), and coreference resolution (`WSC`). In the results, the fine-tuning tasks are summarized as `GLUE`. See Appendix E.3 for fine-tuning parameters.

[3]The tags (e.g., 900 M) refer to the number of pretrained words, not model size.

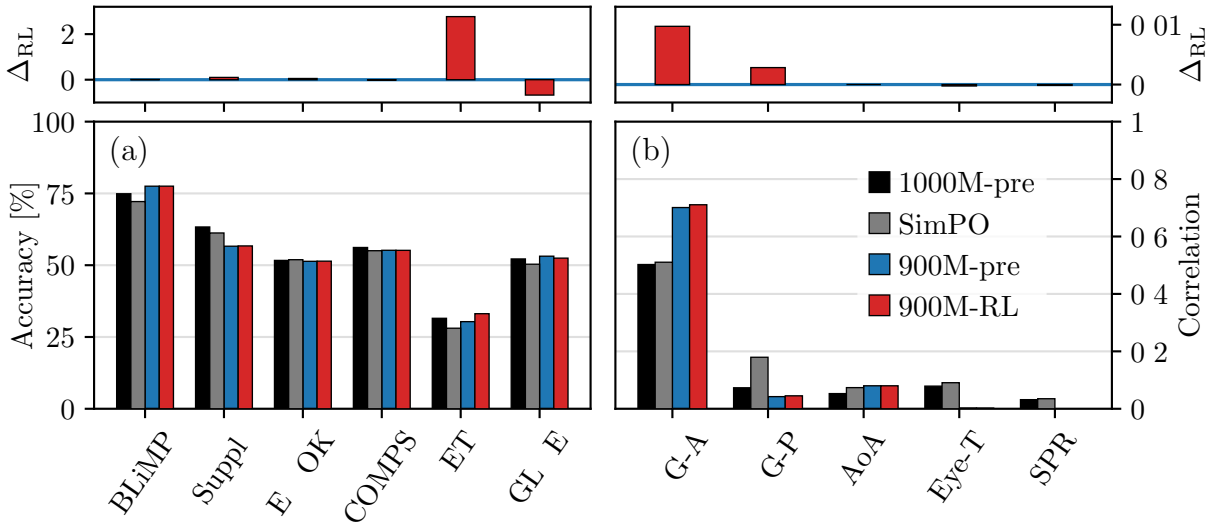


Figure 2: Evaluation on the BabyLM tasks, cf. Table 1. The bottom panels show the (a) accuracy and (b) correlation or partial correlation on the respective tasks for the next-word prediction baseline `1000M-pre`, the interaction baseline `SimPO`, and the model before and after interactive reinforcement learning (RL). The top panels indicate the difference of the 900 M model before and after interaction learning (in percentage points on the left, correlation differences on the right). `GLUE` encompasses all fine-tuning tasks.

5 Results and discussion

We first examine the effect of our interaction model on formal linguistic competence as assessed by the BabyLM evaluation pipeline. We then analyze how storytelling skills improve through reinforcement learning, and explore the training dynamics.

5.1 Formal linguistic competence

We evaluate formal linguistic competence using the BabyLM tasks, comparing our model pretrained on 900 M words before (`900M-pre`) and after (`900M-RL`) interactive reinforcement learning. We also compare with a baseline pretrained on 1,000 M words (`1000M-pre`), and an interaction baseline with a different training objective (`SimPO`). The results are summarized in Fig. 2; for detailed values, see Table 8.

We observe that the two baselines, `1000M-pre` and `900M-pre`, achieve similar performance on most tasks. This suggests that the missing 10% of the pretraining corpus and thus 100 M additional words in pretraining have little effect on formal linguistic competence.

Strikingly, as shown in the top panels in Fig. 2, the accuracy on entity tracking (`ET`) increases the most, from 30.3 % to 33.1 %, and correlations on the two `WUG` tasks improve marginally. Although the teacher reward was not tailored to any of these tasks, improved entity tracking likely reflects the importance of maintaining narrative coherence—specifically, keeping track of characters and objects—in storytelling. Accuracy on the `GLUE` benchmark drops slightly by 0.7 percentage points after interaction. Notably, interactive reinforcement learning does not affect most other BabyLM tasks.

The `SimPO` baseline, despite being exposed to more words during interaction, does not differ much from the baselines and performs slightly worse than `1000M-pre` on `BLiMP`, `BLiMP Supplement`, `ET`, and `GLUE`.

As shown in panel (b) the metrics measuring alignment with psycholinguistic data (`AoA`, `Eye-T`, and `SPR`) have less consistent trends than the accuracy-based scores in panel (a) and the correlations of all models are below 0.1. The `WUG-A` task has a high correlation between 0.5 and 0.7 for all models.

In summary, two observations stand out: First, omitting 10 % of training data (`900M-pre` vs. `1000M-pre`) does not significantly affect the performance on the formal linguistic competence captured by the BabyLM tasks. Second, adding only 1 M additional words of interactive reinforcement learning after pretraining maintains those competences and even improves entity tracking.

5.2 Storytelling

How does interactive learning affect the learning dynamics of a small language model? We first

explore the storytelling performance itself and the data efficiency of the learning setup. Next, we dive deeper into the learning dynamics of the individual storytelling criteria, the influence of the number of pretraining words and the interaction progress.

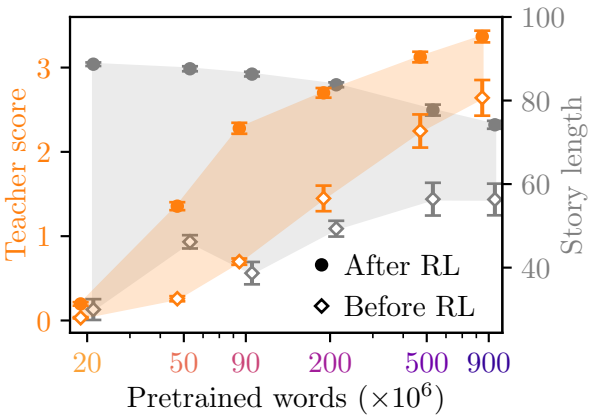


Figure 3: The effect of interactive reinforcement learning (RL) for models with increasing number of pretraining words on two variables: average teacher score (orange, left axis) and story length (gray, right axis). Error bars indicate the standard deviation of the first and last 20 batch averages, respectively. Orange and gray data points are slightly offset horizontally to avoid overlap.

Storytelling skills As shown in Figure 3, after the reinforcement learning (RL) interaction phase, the student models produce stories that are both longer and rated higher by the teacher. At first glance, this indicates that the models successfully learn to optimize the reward, which combines the teacher score and a bonus for story length. However, the extent of the improvement depends strongly on the amount of pretraining.

Specifically, models with more pretraining produce higher-scoring stories, both before and after RL: The 20 M model initially produces short and after RL long stories that the teacher scores almost zero throughout. In contrast, the 90 M and 200 M models show the greatest increase in teacher score, while the most pretrained model, 900 M, gains less from RL, although it ultimately achieves the highest absolute scores. Interestingly, the 900 M model also produces the shortest stories after RL, despite earning the highest ratings, which suggests that it relies least on story length as a shortcut.

In Appendix C, we provide a random sample of stories from the first, middle, and last third of interactions, as well as the best story, for the 90 M and 900 M models. The anthology of all stories produced by the models is available as a Hugging Face dataset.

Data efficiency We find that interactive learning is remarkably data efficient: After RL, the 90 M model receives an average teacher score of 2.3 that outperforms that of the 500 M model before storytelling interaction. Thus, 1 M words of interactive learning achieve the same improvement as 410 M extra words in pretraining. This result aligns with the findings of Wu et al. (2025) and Zhao et al. (2023), who demonstrate that LLMs learn with surprising efficiency in reinforcement learning. This robustness to sparse reward signals—such as the fixed student input in our setup—can be attributed to knowledge of the target domain acquired through sufficient pretraining. In our case, this finding agrees with our observation that a certain amount of pretraining is required before reinforcement learning can meaningfully enhance storytelling skills.

Story quality We analyze the distribution of teacher scores across the criteria used for evaluating the student model's stories. Figure 4 (a) shows the evolution of each criterion's score with the number of interactions for the six models with different amounts of pretraining. To emphasize underlying trends and filter out high-frequency fluctuations of the data, we apply a Gaussian filter.

Overall, the scores for all three criteria increase over time. As illustrated in Fig. 4 (b), models with more pretraining perform better on all criteria. Notably, readability emerges as the hardest criterion, for which even the 900 M model rarely attains two points, while performance in creativity and narrative coherence is substantially better across all models. The limited improvements on readability, which reflects superficial fluency, fit the observation from Section 5.1 that, for example, grammatical knowledge (as measured by BLiMP) is not much affected by the interactive RL, but creativity and coherence improve instead.

Fig. 4 also shows that the 20 M model fails to achieve higher teacher scores except for a minor gain in creativity. Models pretrained for 90 M and 200 M words gain the most on all criteria, whereas more pretraining leads to diminishing returns in teacher scores.[4]

[4]Considering the entropy per word, see Appendix B, we find that it is dominated by the amount of pretraining, with little change during interactive RL. This indicates that improvements in storytelling cannot simply be attributed to changes in output diversity.

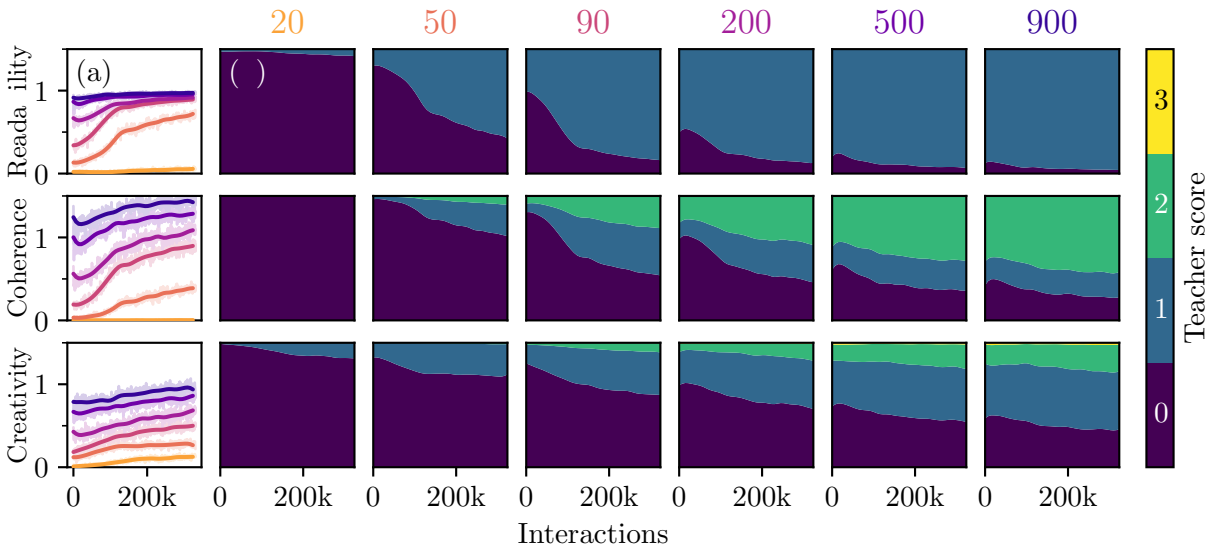


Figure 4: Teacher score over the course of RL interaction for models with increasing pretraining: 20 M, 50 M, . . . , 900 M words. (a) Teacher scores by criterion. Shaded regions show averages per batch, solid lines are Gaussian-smoothed batch averages. (b) Distribution of teacher scores over time.

Learning dynamics Figure 5 (a) illustrates the evolution of teacher score, story length, and KL divergence over the number of interactions. Across all models, both teacher score and story length increase most rapidly until 100k interactions, after which improvements continue but at a slower pace. This deceleration is also reflected in Fig. 4. KL divergence, which quantifies the similarity of the RL-trained model to its pretrained baseline, increases during early training and then stays constant around $\mathrm{KL} = 6$, a convergence determined by the adaptive KL scheduling of PPO. Deviating from this plateau would compromise the total reward signal, thus constraining policy updates. Notably, models with more pretraining, like 500 M and 900 M words, exhibit a decrease in story length after KL convergence before increasing again, potentially signaling a delayed adaptation of the model's reward prediction as these models adjust to changes in the slope of KL divergence.

Fig. 5 (b) combines the trajectories of the different models along three dimensions: story length, teacher score, and number of interactions. These trajectories define a surface, which we approximate with a one-dimensional linear interpolation (surface with blue to yellow gradient). The upper two diagrams in panel (a) correspond to projections of the trajectories, connecting the nonlinear effect of pretraining on the evolution of these variables.

Fig. 5 (c) completes the picture with a projection onto the plane of teacher score and story length, collapsing the dimension of interactions. This view reveals how models with different pretraining navigate the trade-off between story length and teacher score. The 20 M model shows a limited slope, improving primarily in story length. This indicates a threshold: models pretrained on fewer than 50 M words cannot leverage interactive feedback, which implies that some amount of pretraining is necessary for a viable reward signal. In contrast, the 90 M and 200 M models exhibit pronounced improvement in both dimensions. Models with even more pretraining like 500 M and 900 M display diminishing returns, consistent with Fig. 3. Overall, the 90 M model benefits most from interactive learning.

6 Conclusion

Our experiments demonstrate that interactive feedback is highly data efficient for storytelling: With just 1 M words of additional input, storytelling skills reach the equivalent of an additional 410 M words of next-word prediction in pretraining. This result highlights the data inefficiency of next-word prediction and might explain why children acquire language with far less input than today's large language models.

We find that interactive reinforcement learning primarily enhances narrative coherence and creativity, while leaving surface-level fluency—measured by the BabyLM tasks—largely unchanged. An improvement in entity tracking aligns with the training objective focused on storytelling.

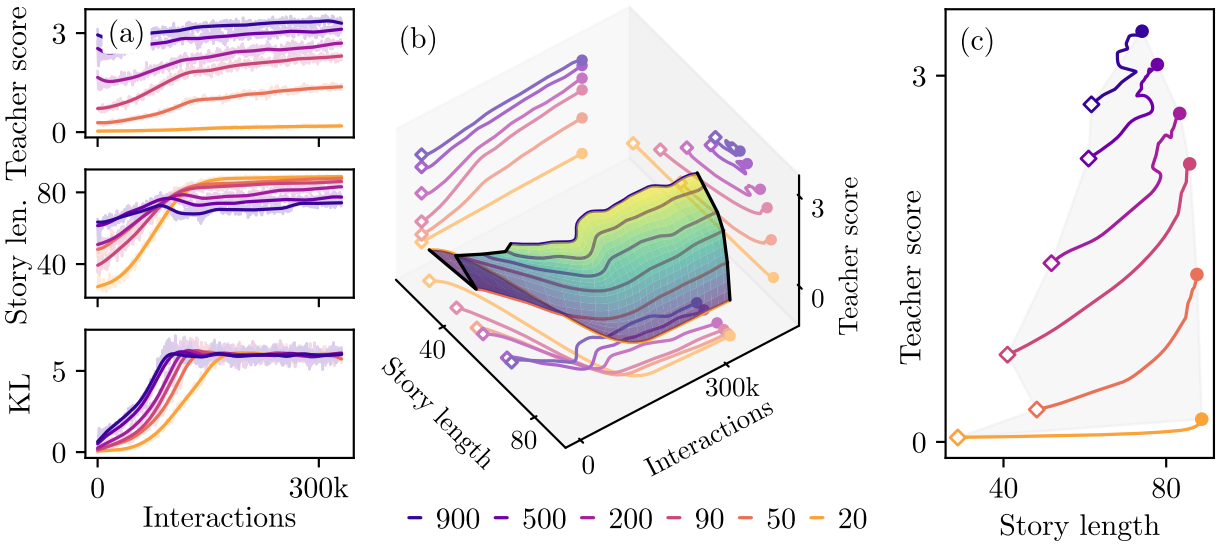


Figure 5: Learning dynamics of the reinforcement learning (RL). (a) Teacher score, story length, and KL divergence by interaction number. The shading shows the average per batch, the solid lines are Gaussian-smoothed batch averages. (b) Training trajectories visualized as a manifold with projections in the dimensions of story length (0 to 90 words) and teacher score (0 to 9 points) and number of interactions. (c) Trajectories in the phase space of teacher score and story length.

Our analysis reveals that models with less pretraining tend to exploit story length as a shortcut, whereas those with 90 M and 200 M words of pretraining benefit the most from interactive learning. Models with more pretraining suffer from diminishing returns from interaction. Notably, we identify a threshold: between 20 M and 50 M words of pretraining are necessary for the model to benefit from interactive reinforcement learning. Examining the nature of this threshold and its parallels to language acquisition in children presents an intriguing avenue for future research.

Limitations

While storytelling RL is highly data efficient, it is by no means computationally efficient: RL on 1 M input words took 20 GPU hours per model, because it involves generating 20 M words of student output for the stories. For comparison, 900 M words of pretraining amounted to less than 10 GPU hours.

Moreover, our analysis focuses on the learning dynamics. We leave a detailed study of the student stories—how content, register, vocabulary, and syntax evolve through interaction—for future work. Mechanistic interpretability methods could also provide insights into how training affects internal model representations.

Furthermore, we weight the three evaluation criteria of the teacher equally, but these weights can be adapted during RL to implement a form of curriculum learning.

Our teacher rewards serve as a heuristic for story quality. Further validation using benchmarks like `OpenMEVA` (Guan et al., 2021) or human annotations would strengthen this approach.

We used a fixed input for story generation, but more diverse corpora (e.g., BabyLM (Charpentier et al., 2025), TinyStories (Eldan and Li, 2023), or WritingPrompts (Fan et al., 2018)) could affect learning outcomes; each with its own tradeoffs regarding narrative content and diversity.

Ethics statement

Importantly, computational language models are not faithful representations of human cognition and should not be anthropomorphized. Rather, they are tools for informing hypotheses about language learning, which should ultimately be tested on human studies.

While the BabyLM challenge targets more sustainable training regimes, model development still requires considerable computing resources. Model development and final training took about 140 kcore-hours in total. Pretraining took 2 hours on 4 A100 GPUs. RL learning took 20 hours on 1 A100 GPU for each of the six RL models (5 - 10 kcore-hours per model).

Acknowledgements

We thank the reviewers for their input. We thank Eva Beck for helpful discussions. Lisa Beinborn's research is partially supported by an *Impulsprofessur* grant from the *zukunft.niedersachsen* program and by a VENI grant (Vl.Veni.211C.039) from the Dutch National Science Organisation (NWO). The authors gratefully acknowledge computing time provided to them at the GWDG HPC cluster.

References

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. 2024. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121.

Luisa Bentivogli, Ido Kalman Dagan, Hoa Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC 2009 Workshop*.

Timothée Bernard, Timothee Mickus, and Hiroya Takamura. 2024. The emergence of high-level semantics in a signaling game. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 200–211, Mexico City, Mexico. Association for Computational Linguistics (ACL).

BNC Consortium. 2007. British National Corpus, XML edition.

Brendon Boldt and David Mortensen. 2024. A review of the applications of deep learning-based emergent communication. arXiv:2407.03302. *Transactions on Machine Learning Research*, arXiv:2407.03302.

Jerome Bruner. 1991. The narrative construction of reality. *Critical Inquiry*, 18(1):1–21.

Joseph Campbell. 2008. *The Hero with a Thousand Faces*, 3rd edition. Bollingen series XVII. New World Library, Novato, Calif.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM turns 3: Call for papers for the 2025 BabyLM workshop. arXiv:2502.10645. *arXiv preprint*.

Gregory Cheatham, Margarita Jimenez-Silva, and Hyejin Park. 2015. Teacher feedback to support oral language learning for young dual language learners. *Early Child Development and Care*, 185:1452–1463.

Cyril Chhun, Pierre Colombo, Fabian M. Suchanek, and Chloé Clavel. 2022. Of human criteria and automatic metrics: A benchmark of the evaluation of story generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5794–5836, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Andy Clark. 2013. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3):181–204.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics (ACL).

Eve V. Clark. 2018. Conversation and language acquisition: A pragmatic approach. *Language Learning and Development*, 14(3):170–185.

Herbert H. Clark and Susan E. Brennan. 1991. Grounding in communication. In Lauren B. Resnick, Levine John M., and Stephanie D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 127–149. American Psychological Association, Washington.

Herbert H. Clark and Edward F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13(2):259–294.

Sayantan Dasgupta, Trevor Cohn, and Timothy Baldwin. 2023. Cost-effective distillation of large language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7346–7354, Toronto, Canada. Association for Computational Linguistics (ACL).

Andrea Gregor De Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 English sentences, aligned with existing EEG and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

DeepSeek-AI. 2025. DeepSeek-V3 technical report. arXiv:2412.19437. *arXiv preprint*.

Richard Diehl Martinez, Zébulon Goriely, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. Mitigating frequency bias and anisotropy in language model pre-training with syntactic smoothing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5999–6011, Miami, Florida, USA. Association for Computational Linguistics (ACL).

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases.

In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How small can language models be and still speak coherent English? arXiv:2305.07759. *arXiv preprint*.

Linnea Evanson, Yair Lakretz, and Jean Rémi King. 2023. Language acquisition: Do children and language models follow similar learning stages? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12205–12218, Toronto, Canada. Association for Computational Linguistics (ACL).

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics (ACL).

Martin Gerlach and Francesc Font-Clos. 2020. A standardized Project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, D. Kimbrough Oller, John H. L. Hansen, and Terrance D. Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

Nathan Godey, Éric Clergerie, and Benoît Sagot. 2024. Anisotropy is inherent to self-attention in transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 35–48, St. Julian's, Malta. Association for Computational Linguistics.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520 vol. 1, San Francisco, CA, USA. IEEE.

Jonathan Gottschall. 2012. *The Storytelling Animal: How Stories Make Us Human*. Houghton Mifflin Harcourt, Boston.

Aaron et. al. Grattafiori. 2024. The Llama 3 herd of models. arXiv:2407.21783. *arXiv preprint*.

Jian Guan, Zhexin Zhang, Zhuoer Feng, Zitao Liu, Wenbiao Ding, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2021. OpenMEVA: A benchmark for evaluating open-ended story generation metrics. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6394–6407, Online. Association for Computational Linguistics (ACL).

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. arXiv:2403.04642. *arXiv preprint*.

Valentin Hofmann, Leonie Weissweiler, David Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics (ACL).

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of world knowledge (EWoK): A cognition-inspired framework for evaluating basic world knowledge in language models. arXiv:2405.09605. *arXiv preprint*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics (ACL).

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics (ACL).

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. 2020. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7663–7674, Online. Association for Computational Linguistics (ACL).

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online. Association for Computational Linguistics (ACL).

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Brian MacWhinney. 2014. *The CHILDES Project: Tools for Analyzing Talk, Volume II: The Database*, 3rd edition. Psychology Press, New York.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics (ACL).

Mitja Nikolaus and Abdellah Fourtassi. 2021. Modeling the interaction between perception-based and production-based learning in children's early acquisition of semantic knowledge. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 391–407, Online. Association for Computational Linguistics (ACL).

Mitja Nikolaus and Abdellah Fourtassi. 2023. Communicative feedback in language acquisition. *New Ideas in Psychology*, 68:100985.

Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. 2024. The ultimate guide to fine-tuning LLMs from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. arXiv:2408.13296. *arXiv preprint*.

Rachel Ryskin and Mante S. Nieuwland. 2023. Prediction during language comprehension: What is next? *Trends in Cognitive Sciences*, 27(11):1032–1052.

Jenny R. Saffran. 2020. Statistical language learning in infancy. *Child Development Perspectives*, 14(1):49–54.

Swarnadeep Saha, Peter Hase, and Mohit Bansal. 2023. Can language models teach weaker agents? Teacher explanations improve students via personalization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 62869–62891, Red Hook, NY, USA. Curran Associates Inc.

Muskan Saraf, Sajjad Rezvani Boroujeni, Justin Beaudry, Hossein Abedi, and Tom Bush. 2025. Quantifying label-induced bias in large language model self- and cross-evaluations. arXiv.2508.21164. *arXiv preprint*.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics (ACL).

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Lennart Stöpler, Rufat Asadli, Mitja Nikolaus, Ryan Cotterell, and Alex Warstadt. 2025. Towards developmentally plausible rewards: Communicative success as a learning signal for interactive language models. arXiv:2505.05970. *arXiv preprint*.

Michael Tomasello. 2008. *Origins of Human Communication*. The MIT Press.

Michael Tomasello. 2014. The ultra-social animal. *European Journal of Social Psychology*, 44(3):187–194.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics (ACL).

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language*

Processing, pages 6508–6524, Singapore. Association for Computational Linguistics (ACL).

Ethan Gotlieb Wilcox, Michael Y. Hu, Aaron Mueller, Alex Warstadt, Leshem Choshen, Chengxu Zhuang, Adina Williams, Ryan Cotterell, and Tal Linzen. 2025. Bigger is not always better: The importance of human-scale language modeling for psycholinguistics. *Journal of Memory and Language*, 144:104650.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics (ACL).

Haoze Wu, Cheng Wang, Wenshuo Zhao, and Junxian He. 2025. Model-task alignment drives distinct RL outcomes. arXiv.2508.21188. *arXiv preprint*.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. 2023. The next chapter: A study of large language models in storytelling. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 323–351, Prague, Czechia. Association for Computational Linguistics (ACL).

Xingmeng Zhao, Tongnian Wang, Sheri Osborn, and Anthony Rios. 2023. BabyStories: Can reinforcement learning teach baby language models to write better stories? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 186–197, Singapore. Association for Computational Linguistics (ACL).

A Evaluation

We use the evaluation pipeline of the 2025 BabyLM Challenge (Charpentier et al., 2025). In Table 1, we provide an overview of the evaluation data.

B Entropy

Figure Fig. 6 shows that the average entropy per word increases slightly at the beginning of training, staying mostly constant until the end of reinforcement learning, but the entropy is otherwise not substantially correlated with story length or teacher score. Pretraining, on the other hand, has a strong influence on the entropy per word.

C Sample stories

Best story by reward and example stories—randomly sampled from the first, second, and last third of RL training—are listed in Table 2 for 90 M pretrained words and Table 3 for 900 M pretrained words. Interestingly, the best story for 900 M pretrained words is a meta-story—a story about a story—that directly appeals to the teacher evaluation by describing a "great story". Each model produced about 20 M words during RL training, which amounts to about 50,000 pages. The full anthology is available as a Hugging Face dataset.

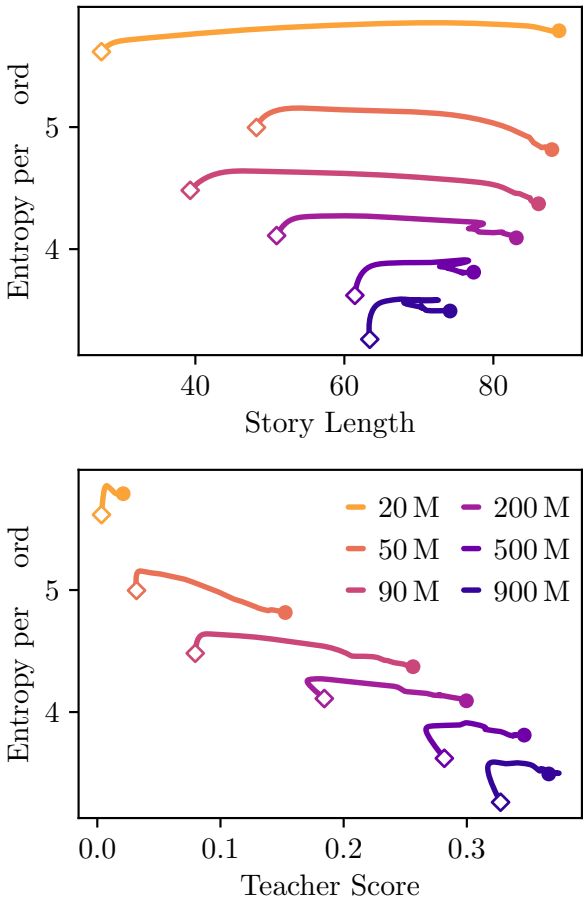


Figure 6: Entropy per word by story length and teacher score during interactive RL for different amounts of pretrained words. An empty diamond marks the start of a trajectory, a filled circle the end.

Setting	Dataset	Prediction Task	Evaluation Metric	Reference
Zero-shot	BLiMP	Grammatical acceptability	Accuracy	Warstadt et al. (2020)
	Suppl.	Discourse acceptability	Accuracy	Warstadt et al. (2023)
	EWOK	Conceptual knowledge	Accuracy	Ivanova et al. (2025)
	COMPS	Property knowledge	Accuracy	Misra et al. (2023)
	WUG-A	Morphol. generalization (adj.)	Spearman's ρ	Weissweiler et al. (2023)
	WUG-P	Morphol. generalization (verbs)	Spearman's ρ	Hofmann et al. (2025)
	AoA	Age of acquisition	Pearson's ρ	Chang and Bergen (2022)
	Eye-T	Eye-tracking fixations	Squ. partial corr. pR^2	De Varda et al. (2024)
	SPR	Reading times	Squ. partial corr. pR^2	De Varda et al. (2024)
	ET	Entity Tracking	Accuracy	Kim and Schuster (2023)
Fine-tuning	BoolQ	Question answering	Accuracy	Clark et al. (2019)
	MultiRC	Question answering	Accuracy	Khashabi et al. (2018)
	MNLI	Natural language inference	Accuracy	Williams et al. (2018)
	RTE	Entailment	Accuracy	Bentivogli et al. (2009)
	MRPC	Paragraph identification	Accuracy	Dolan and Brockett (2005)
	QQP	Question similarity	Accuracy	Iyer et al., (2017)
	WSC	Coreference resolution	Accuracy	Levesque et al. (2012)

Table 1: Overview of evaluation datasets in the BabyLM pipeline.

(1, 2, 3)	**Length:** 82	**Reward:** 0.71	**#** 318,619

when it was given to the sun. Did he tell you that I wrote a letter, then that the moon was not frightened of telling her story? She introduced her red coat into the eyes of his safety. Perhaps she loved that, though she had seen sad old laughter. She now told her primary harmon they in spite of the name as she had never seen a star light thunder escort. She found herself there a dragon doing her best at it

(0, 0, 0)	**Length:** 25	**Reward:** 0.07	**#** 43,434

she will possess me a descend-weller repwined feast. Of this of the Saxon divinity and for fifty months will occur; " Mr. March reppainted .</s>

(1, 1, 1)	**Length:** 67	**Reward:** 0.43	**#** 217,857

hitherto hathine before twenty-three of people had lost their Senator Ragnzan leave in the decision, formally on the fact that he would make-up payment. After Commonwealth negotiated against a white elector offenders, Captain Spndez Junpped with the alliance and denied the suggestion which was only a vote from him: he seems to men such as England and Wales, but he said not to have been a party,

(1, 2, 1)	**Length:** 71	**Reward:** 0.52	**#** 321,361

which whenever it may not be trusted. 'Cause she really would feel about that she might have recognized her lyrics. I encourage her. The love is not yet permitted to pronouncemy to the contrary." She was in a state that it was a small, sweet gift. She kept pieces of the discussion, and she heard her dear face. Her newence was based upon her key to the audit that she had

Table 2: Examples of generated stories from the 90 M model: overall best story (top) ranked by teacher and length reward, plus random samples (seed 42) from the first, second, and last third. Prompt: "*Let me tell you a long, magical tale. Once upon a time, in a faraway land*".

(2, 3, 3)	**Length:** 76	**Reward:** 0.85	**#** 8,821

everything seemed as if it were a fairy tale—so much as you have told us a great story that we cannot bear to believe. There was something, however, very different from the description of the whole story, most surprising. If the story reached out, then, and it was always the story at the end, it was always something inexpressibly a story that made a great deal of sense. But it was all a mistake and wonder

(1, 2, 2)	**Length:** 76	**Reward:** 0.61	**#** 43,434

there lived old men like me, the flowers of Jed serene rose from every mountain and valley, and they grew wildly from one to another. The inhabitants of those under one tree and the branches in the other, had been bewitched in their own world by their native dwelling; but having enabled them to approach them they had for good to take an active place among mankind. They resided many weary people, but had always a

(1, 0, 0)	**Length:** 63	**Reward:** 0.26	**#** 217,857

there lived an unquenchable king. But cannot the words wear off, and for ten days he was forced to stake his head on one native bird-cage, while the hunters came in for us the fish-skin preservation. He was a dread of poor little war-birds, and a more likeable wickedness so lonesome in proportion to his cruel fangs as a young bird devouring li

(1, 0, 0)	**Length:** 67	**Reward:** 0.27	**#** 321,361

immature and upland boy, he was dazzled by the tremendous overlooks of his race had lighted. He met a spirit who had been there all the day to bespeak in the midst of many years, and answered: "Hear him, Don Carlos, from there he lent it to reality; He is a different kind of drunken-looking man; I consider him much creamered after his teeth." In the same

Table 3: Examples of generated stories from the 900 M model: overall best story (top) ranked by teacher and length reward, plus random samples (seed 42) from the first, second, and last third.

D Evaluation instructions

```
You are a helpful teacher grading a student story. Be nice!
Only evaluate the student story itself, not the story prompt.
Given the student's word limit of about 80 words,
score the story on each of these three categories separately
on a scale from 0 to 3,
where 0 is the worst and 3 is the best.

Readability:
0 - Frequent and severe grammar errors; difficult to understand.
1 - Noticeable grammar errors; mostly understandable.
2 - Few minor grammar errors; well-formed overall.
3 - Correct grammar; well written.

Narrative Coherence:
0 - No story: completely incoherent or too short.
1 - No logical flow, confusing narrative.
2 - Mostly coherent story and not cut off.
3 - Coherent and logically structured story.

Creativity:
0 - Dull or incomprehensible.
1 - Somewhat creative; mostly predictable.
2 - Fairly creative and engaging.
3 - Highly original, imaginative, and engaging.

If the student story is empty ("") or less than a full sentence,
you must give the score 0 0 0!

Provide your scores, separated by single spaces, in the format:
Readability, Narrative, Creativity = _ _ _

Respond ONLY with this sequence of three numbers
without any extra text or explanation.

Story Prompt:
`{{story_prompt}}`

Student Story:
"`{{student_completion}}`"

Readability, Narrative, Creativity =
```

Source	Ratio	Domain	Reference
BNC	8%	Dialogue	BNC Consortium (2007)
CHILDES	29%	Dialogue, child-directed	MacWhinney (2014)
Proj. Gutenberg	26%	Fiction, nonfiction	Gerlach and Font-Clos (2020)
OpenSubtitles	20%	Dialogue, scripted	Lison and Tiedemann (2016)
Simple Eng. Wiki.	15%	Nonfiction	–
Switchboard	1%	Dialogue	Godfrey et al. (1992), Stolcke et al. (2000)

Table 4: Composition of the BabyLM corpus.

E Model parameters

BabyLM corpus The composition of the BabyLM corpus is listed in Table 4. It comprises 100 M words, of which we use 90% for pretraining and tokenization.

E.1 Pretraining

Model and training The model parameters are listed in Table 5. The vocab size of the tokenizer is 16,000 to match the baseline `1000M-pre` and the interaction baseline `SimPO`, which have vocab size 16,384. We use different values for seed, batch size, gradient accumulation, and learning rate compared with the baselines.

Hyperparameter	Value
Number of epochs	10
Context length	512
Batch size	16
Gradient accum. steps	4
Learning rate	0.0005
Number of steps	211,650
Warmup steps	2,116
Gradient clipping	1
Seed	42
Optimizer	AdamW
Optimizer β_1	0.9
Optimizer β_2	0.999
Optimizer ε	10^{-8}
Tokenizer	ByteLevelBPE
Tokenizer vocab size	16,000
Tokenizer min. frequency	2

Table 5: Hyperparameters used for pretraining.

E.2 Reinforcement learning

See Table 6.

Parameter	Value
Student context length	512
Seed	42
Batch size	360
Student sampling temp.	1
Top k	0
Top p	1
Max. new tokens (student)	90
Teacher model	Llama 3.1 8B Instr.
Teacher context length	1,024
Student sampling temp.	0.2
Max. new tokens (teacher)	6
Gradient acc. steps	1
Adapt. KL control	True
Init. KL coef.	0.2
Learning rate	1×10^{-6}
Student input limit	1 M words

Table 6: PPO Training Hyperparameters. Other parameters defaults of TRL 0.9.4.

E.3 Fine-tuning

See Table 7.

Hyperparameter	Value
Number of Epochs	10
Batch Size	16
Learning Rate	3×10^{-5}
Warmup percentage	6 %
Optimizer	AdamW
Weight decay	0.01
Scheduler	cosine
Dropout	0.1

Table 7: Hyperparameters used for fine-tuning.

F BabyLM evaluation results

See Table 8.

Task	1000M-pre	SimPO	900M-pre	900M-RL
BLiMP	74.88	72.16	77.52	**77.53**
Suppl.	**63.32**	61.22	56.62	56.72
EWOK	51.67	**51.92**	51.36	51.41
COMPS	**56.17**	55.05	55.20	55.18
ET	31.51	28.06	30.34	**33.11**
GLUE	52.18	50.35	**53.14**	52.46

Task	1000M-pre	SimPO	900M-pre	900M-RL
WUG-A	0.502	0.510	0.701	**0.711**
WUG-P	0.073	**0.179**	0.042	0.045
AoA	0.053	0.074	**0.080**	**0.080**
Eye-T	0.079	**0.091**	0.003	0.002
SPR	0.032	**0.035**	0.000	0.000

Table 8: BabyLM task scores for the four models from Fig. 2. Accuracy metrics are reported as percentages, WUG-A/P as Spearman's ρ, AoA as Pearson's ρ, Eye-T and SPR as partial correlations $\mathrm{p}R^2$. Bold indicates the best model for each task.

You are an LLM teaching a smaller model everything you know: Multi-task pretraining of language models with LLM-designed study plans

Wiktor Kamzela[1] **and Mateusz Lango**[1,2] **and Ondřej Dušek**[2]
[1]Poznan University of Technology, Faculty of Computing and Telecommunications, Poznan, Poland
[2]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
wiktor.kamzela@student.put.edu.pl, {lango,odusek}@ufal.mff.cuni.cz

Abstract

This paper proposes a multi-task pre-training of language models without any text corpora. The method leverages an existing Large Language Model (LLM) to generate a diverse corpus containing training data for 56 automatically designed tasks and uses generated labels to enhance the training signal. The method does not rely on hidden states or even output distributions of the teacher model, so may be employed in scenarios when the teacher LLM is available only through an API. The conducted experiments show that models trained on the proposed synthetic corpora achieve competitive or superior performance compared to those trained on same-sized human-written texts.

1 Introduction

Pretraining of language models (LMs) typically relies on massive text corpora collected from the web, books, and other sources (Gao et al., 2020; Bai et al., 2023). While this paradigm has proven highly effective for building large language models (LLMs), it also poses a significant challenge: training requires enormous computational resources to process large datasets. This limitation has sparked research interest in approaches that reduce data requirements, such as training models on smaller corpora (Hu et al., 2024) or leveraging knowledge distillation from already trained, larger models (Gu et al., 2024). Knowledge distillation, however, typically assumes access to the teacher model's hidden states, parameter values, or output distributions, which is rarely possible when the model is exposed only through an API (Xu et al., 2024).

A parallel line of research has explored the use of LLMs to generate synthetic data for model fine-tuning. Prior work has shown promising results in tasks such as text classification (Li et al., 2023), data augmentation (Long et al., 2024), and instruction tuning (Li et al., 2024). To the best of our knowledge, however, synthetic data generation has not yet been applied to *pretraining* language models. This raises two key challenges. First, LLMs tend to produce similar outputs from the same data generation prompt, making it difficult to obtain the level of diversity required for pretraining. Second, achieving strong performance on small datasets requires more efficient training techniques.

In this paper, we address these challenges by proposing multi-task pre-training of language models using an LLM-designed study plan – synthetic data that is not only automatically generated, but also composed of tasks picked by a teacher LLM. First, we instruct a teacher LLM to design a *study plan* for a smaller model, with the goal of teaching the smaller model how to solve all NLP tasks that an LLM should be able to handle. We then let the LLM iteratively generate a *dataset for each task* indicated in the previous step. This task-oriented approach to synthetic data generation, combined with the additional prompt extension strategies proposed, enhances the diversity of the output data and provides multiple synthetic labels for each text. The generated labels provide an opportunity to enrich the training signal for the language model through our proposed *multi-task loss*, which, in addition to the standard masked language modelling (MLM) objective, incorporates multiple text classification and sequence tagging losses.

The experimental evaluation performed on SuperGLUE (Wang et al., 2019) and BLiMP (Warstadt et al., 2020) benchmarks indicates that language models pretrained on synthetic data generated by the proposed technique perform competitively compared to models trained using human-written texts of the same size. Our models obtain the best average performance across both benchmarks among models trained on small corpora of 1M words. For 10M-word training corpora, our models perform best on fine-tuned downstream tasks of SuperGLUE, while models train on human-written data are better on BLiMP.

Proceedings of the First BabyLM Workshop, pages 481–499
November 8, 2025

This paper describes our submission to the interaction track of BabyLM Challenge 2025 (Charpentier et al., 2025). The model pretrained on a small 1M words multi-task corpora is publicly available at https://huggingface.co/Wector1/Multitask-pretraining_1M.

2 Problem statement

The goal of the presented method is to train a language model without relying on any preexisting text corpora. Instead, a selected large language model (LLM) is used as a teacher, but its weights, hidden states, or output distributions are not revealed to the student model. The teacher model therefore generates synthetic training data, which is then used to train the student model.

3 Task-oriented data generation

Our data generation pipeline consists of two fully automatic stages: (1) *study plan design* (selection of target NLP tasks) and (2) *generation of training examples* for each training task. In the study plan step, a teacher LLM enumerates desirable NLP tasks, designs the corresponding annotation schemas (i.e., list of target classes/tags) and constructs prompts that will generate training data following the schemas. The example generation step runs the provided prompts and diversifies them by adding requests to generate examples of a given class, a given difficulty level, or containing selected words. The overview of the data generation process is presented in Fig. 1.

3.1 Task generation

The teacher LLM is asked to design a study plan for a smaller LLM to teach the student everything it "knows". The study plan is generated in four iterations, each time asking the teacher to create a study plan for one of the four "lessons" (NLP task types): *text classification*, *text pair classification*, *sequence tagging*, and *text generation*. Apart from instructing LLM to focus on English-only tasks, the prompt requests the teacher to build a diversified list of tasks and to avoid confusion with other task categories. All prompts are provided in App. A.

Next, for each suggested task (other than text generation tasks), the LLM is asked to generate an annotation schema, containing the list of classes and their descriptions. Finally, the teacher is instructed to design a list of prompts that would make an LLM generate a dataset for a given task with the given annotation schema. The final result of the process is a list of tasks with the following attributes: name, description, task type (one of the four listed above), a list of classes/tags (with their definitions), and a list of multiple prompts that can generate a training dataset for said task.

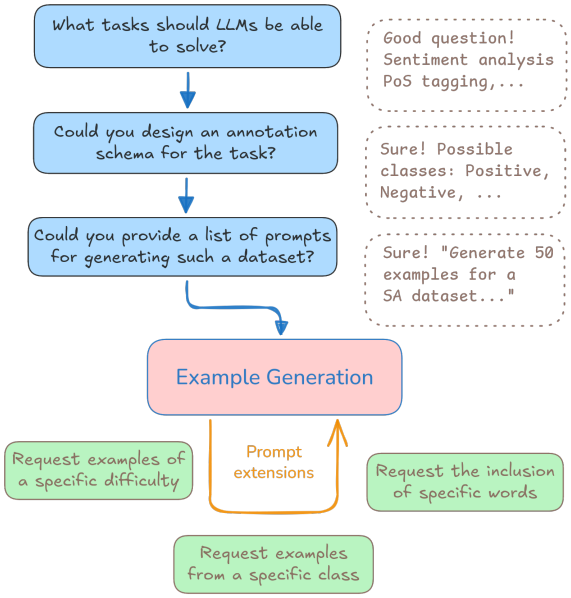


Figure 1: An overview of our data generation strategy for large language model pretraining.

3.2 Example generation

The training examples generation for each task is performed by collecting LLM responses for prompts generated in the previous step. As prompts are designed automatically, to avoid potential confusion during generation, we additionally used a system prompt that contains the task description, input-output specification and the instruction to respond with 50 examples.

A major issue when generating a large dataset with LLM is obtaining diverse examples. To this end, we designed three prompt extenders: *difficulty extender*, *label extender* and *vocabulary extender*. Each extender worked by appending a sentence with additional instructions to the original prompt.

The *difficulty extender* asks for easy, medium and hard to classify examples. The *label extender* specifically requests examples belonging to a single selected class. The *vocabulary extender* is the most advanced: it tracks the vocabulary in already generated examples and requests texts containing at least one of five target words. These words are selected as the least frequent in the already generated samples, except for words occurring less

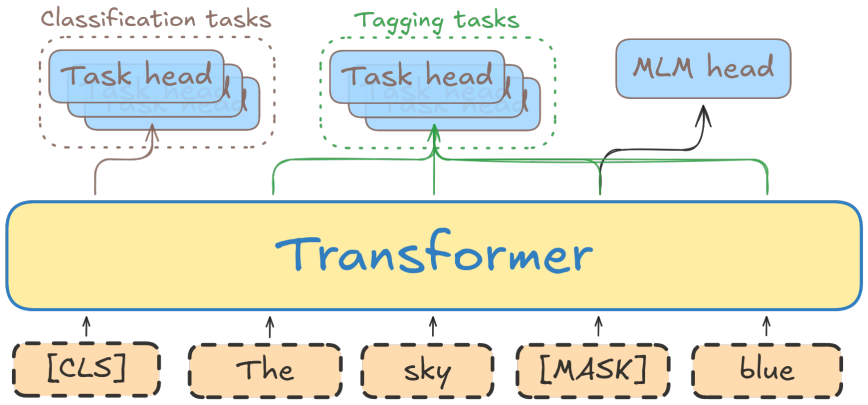


Figure 2: An overview of the proposed architecture of task-augmented pretraining of a language model.

than three times (to avoid noise). This allows for a gradual expansion into more complex vocabulary.

3.3 Dataset postprocessing and multi-task labeling

The data generated from the previous steps is a collection of datasets for different tasks, where each component dataset contains only one type of label. To fully embrace the potential of multi-task pretraining, we generate labels/tags *for all tasks* for each given input. For instance, a given sentence originally generated for the task of spam classification will additionally obtain tags for part-of-speech classification, sentiment analysis, etc. To this end, we asked the teacher LLM to act as a classifier/-tagger for each given task (except text generation tasks) and to provide labels for the whole dataset. See prompt in App. A.

Finally, the generated dataset undergoes a simple filtering, consisting of deduplication and removing all instances that contain fewer than 3 words or contain non-English characters. Additionally, any inconsistent synthetic annotation (e.g., sequence tag lists of incorrect lengths or labels outside of the designed schema) is discarded, i.e., each instance in the final data includes labels for most but not necessarily all NLP tasks.

4 Task-augmented pretraining

To take advantage of the generated preprocessing data that contains labels for artificially constructed tasks, we propose a task-augmented pretraining method that modifies the standard transformer architecture by adding multiple classification/tagging heads. Each task head is associated with a task loss function, which enriches the standard MLM loss with an additional training signal, allowing for training with smaller datasets. The overview of the architecture is presented in Fig. 2.

Language modeling The architecture of our model is a bidirectional transformer with the input format following that of BERT. The input sentence begins with a start token `[CLS]` and finishes with `[SEP]`. If the input was generated from a text pair classification task, the input texts are also separated with `[SEP]` token. Note that in text generation tasks, the sentences are not separated by any special tokens as they are only used for standard MLM objective.

During pretraining, the input to the model is perturbed using default Masked Language Modeling parameters in HuggingFace. More concretely, 15% of input tokens are masked: 80% of them are replaced by a special `[MASK]` token, 10% is replaced with a random token and the remaining 10% is left unchanged. An MLM classification head is attached to the output embedding of each masked word that predicts the word at the given position.

$$\mathcal{L}_{MLM} = -\sum_{x_i \in Masked} \log P(x_i | Masked(x)_i)$$

Task heads For each task (except text generation tasks) present in the dataset, a new classification head is constructed, which takes as input the output of the final layer of the transformer network. For efficiency, the tasks are performed on the same, i.e. masked, input as MLM.

For tagging tasks, the corresponding task head is applied to the representation of every input token.

$$\mathcal{L}_{tag\_task_j} = -\sum_{x_i} \log P(y_i^{(j)} | Masked(x)_i)$$

For text and text pair classification tasks, the classification head is applied to the `[CLS]` token.

$$\mathcal{L}_{class\_task_j} = -\log P(y_i^{(j)}|Masked(x)_{[CLS]})$$

Note that every input has multiple labels corresponding to different tasks, and all classification heads are applied simultaneously. However, in the case of the data generation process failing to generate labels for some tasks, the task heads are dynamically detached from the transformer (i.e., only task heads for which labels are available are used).

Multi-task loss The final loss optimized by the model during pretraining is a weighted sum of masked language modeling loss and task losses.

$$\begin{aligned}\mathcal{L} = \; & \mathcal{L}_{MLM} \\ & + w_J J^{-1} \sum_j \mathcal{L}_{class\_task_j} \\ & + w_K K^{-1} \sum_k \mathcal{L}_{tag\_task_k}\end{aligned}$$

where J is the number of text and text pair classification tasks and K is the number of sequence tagging tasks, w_J and w_K are hyperparameters of the loss function.

5 Experiments

5.1 Experimental setup

Data generation The data generation pipeline was executed with Llama 3.1 8B Instruct (Grattafiori et al., 2024) as the teacher LLM. The pipeline was implemented with vLLM library (Kwon et al., 2023), during the generation repetition penalty was set to 1.1, top_k and top_p parameters to 40 and 0.9, respectively.

To fully automate the data generation process, after generating artifacts such as the list of tasks in free-text, the LLM was asked to reformat its response into JSON, with a structured output format enforced as provided by VLLM library. We find that this two-step generation resulted in more diverse samples than directly enforcing the generation of structured output. This generation strategy was also used for all the described interactions with LLM.

Three versions of the dataset were generated:

- *Multi-task corpus* – corpus containing 56 diverse tasks proposed by the teacher model. The full list of tasks is given in App. B. Some of the tasks designed by the teacher model would most probably not be proposed by a human expert, e.g. a text summarization task which belongs to the text classification category (detect if a given text is a summary) or caption writing for images (our model is text-only). Nevertheless, we decided to keep them, assuming that LLM will be consistent while producing and annotating such datasets, which can provide some additional signal for the student model.

- *Text generation corpus* – corpus constructed by the proposed method, but with tasks limited to the text generation category. To obtain a longer list of tasks, the model was prompted to provide an exhaustive list of topics that should be contained in the training corpora of an LLM. This resulted in 58 text generation tasks (talk and discuss a given topic). The rest of the pipeline (i.e. prompt construction, example generation) was performed as previously described.

- *Vocabulary-controlled corpus* – corpus generated identically as "Text generation corpus", but half of it was generated by the LLM using only 5k different tokens (all other tokens were masked from the prediction head). The idea was that providing a lot of training data on a limited vocabulary would help the model better learn the grammar and the representation of the most frequent words. The tokens were selected by running the tokenizer on the whole English Wikipedia corpus[1].

The datasets were generated in two sizes: 1M and 10M words. The datasets are constructed by selecting 1000 examples (10k for the 10M version) from every tagging/text classification task, and the rest consists of texts from the text generation tasks.

Model architecture Our model architecture is based on ModernBERT (Warner et al., 2025) implementation from HuggingFace library (Wolf et al., 2020). Two model sizes were tested:

- 149M[2] architecture of original ModernBERT-small with default parameters, except for the context size set to 256.

[1] https://github.com/GermanT5/wikipedia2corpus

[2] The model size does not include the size of task heads, as they are discarded after pretraining.

	D. size	Dataset	Epochs	BoolQ	MNLI	MultiRC	RTE	WSC	MRPC	QQP	BLiMP	S. GLUE	Average
ModernBERT 149M	1M	Text gen.	10	0.686	0.452	0.664	0.518	0.635	0.701	0.690	55.22	0.621	58.65
		Multi-task	10	0.713	0.444	0.669	0.554	0.615	0.730	0.715	56.80	0.634	60.11
		Vocab. c.	10	0.689	0.451	0.665	0.532	0.692	0.706	0.692	53.12	0.633	58.19
		Text gen.	100	0.689	0.459	0.666	0.554	0.654	0.706	0.713	55.56	**0.634**	59.50
		Multi-task	100	0.696	0.429	0.667	0.583	0.615	0.730	0.700	**57.82**	0.631	**60.48**
		Vocab. c.	100	0.691	0.448	0.665	0.547	0.635	0.721	0.704	56.09	0.630	59.55
		Text gen.	500	0.698	0.452	0.664	0.540	0.635	0.676	0.731	55.82	0.628	59.31
		Multi-task	500	0.692	0.416	0.660	0.532	0.615	0.716	0.705	57.08	0.619	59.51
		Vocab. c.	500	0.703	0.434	0.675	0.540	0.615	0.750	0.705	55.59	0.632	59.39
	10M	Text gen.	10	0.708	0.494	0.665	0.525	0.654	0.745	0.744	61.84	0.648	63.32
		Multi-task	10	0.701	0.453	0.666	0.576	0.635	0.730	0.729	64.38	0.641	64.25
		Vocab. c.	10	0.704	0.485	0.674	0.568	0.635	0.745	0.740	61.78	0.650	63.39
		Text gen.	50	0.698	0.526	0.670	0.561	0.654	0.730	0.767	63.06	0.658	64.44
		Multi-task	50	0.707	0.445	0.673	0.547	0.615	0.696	0.733	**65.19**	0.631	64.14
		Vocab. c.	50	0.702	0.509	0.673	0.619	0.654	0.735	0.758	63.76	**0.664**	**65.10**
ModernBERT 39M	1M	Text gen.	10	0.680	0.429	0.652	0.525	0.654	0.686	0.709	54.12	0.619	58.03
		Multi-task	10	0.691	0.434	0.653	0.540	0.635	0.755	0.715	56.15	0.632	59.66
		Vocab. c.	10	0.677	0.445	0.665	0.525	0.635	0.706	0.718	52.48	0.624	57.46
		Text gen.	100	0.684	0.458	0.666	0.561	0.654	0.706	0.685	55.87	0.631	59.47
		Multi-task	100	0.687	0.437	0.658	0.532	0.654	0.725	0.710	57.63	0.629	**60.28**
		Vocab. c.	100	0.683	0.440	0.669	0.518	0.673	0.701	0.708	56.42	0.627	59.58
		Text gen.	500	0.683	0.433	0.662	0.532	0.692	0.730	0.720	55.75	**0.636**	59.68
		Multi-task	500	0.680	0.414	0.649	0.583	0.615	0.676	0.717	**58.60**	0.619	60.25
		Vocab. c.	500	0.696	0.438	0.666	0.561	0.635	0.706	0.712	55.25	0.631	59.15
	10M	Text gen.	10	0.686	0.459	0.658	0.525	0.654	0.706	0.727	60.07	0.631	61.57
		Multi-task	10	0.683	0.445	0.665	0.525	0.635	0.721	0.727	63.30	0.629	63.08
		Vocab. c.	10	0.678	0.464	0.666	0.576	0.654	0.701	0.725	59.31	0.638	61.54
		Text gen.	50	0.684	0.514	0.676	0.590	0.654	0.721	0.761	61.45	**0.657**	63.58
		Multi-task	50	0.693	0.454	0.667	0.504	0.596	0.711	0.728	**65.08**	0.622	**63.63**
		Vocab. c.	50	0.686	0.484	0.669	0.561	0.635	0.706	0.747	62.49	0.641	63.30

Table 1: Results of evaluation of trained models on BLiMP and SuperGLUE (S. GLUE) benchmark. The best results for a given model and data size are bolded.

- 39M ModernBERT architecture with halved hidden size to 384, intermediate size to 576, and 16 layers.

Training details Models were trained with AdamW optimizer with 128 batch size. Learning rate followed the cosine schedule with 500 warmup steps and a learning rate of 0.0003. The weights of the multi-task loss (see Sec. 4) were selected to $w_J = w_K = 0.5$. A small weight decay of 0.01 was applied.

The number of epochs depended on the size of the dataset. The smaller 1M dataset was tested with 10 epochs (10M tokens seen during training), 100 epochs (100M tokens) and 500 epochs (500M tokens). The larger 10M dataset was tested with 10 epochs (100M tokens) and 50 epochs (500M tokens).

Evaluation Our evaluation follows the evaluation framework provided by the BabyLM Challenge organizers (Charpentier et al., 2025). More concretely, we evaluated our model's language understanding capabilities using the SuperGLUE benchmark, encompassing the tasks BoolQ, MNLI, MultiRC, RTE, WSC, MRPC, and QQP (Wang et al., 2019). For each task, pretrained models were fine-tuned with default parameters provided by BabyLM organizers without hyperparameter tuning. Additionally, we benchmarked grammatical knowledge using BLiMP (the Benchmark of Linguistic Minimal Pairs), which comprises 67 subdatasets of minimal sentence pairs probing syntax, morphology, and semantics (Warstadt et al., 2020). For the convenience of model comparisons, we also report the average of BLiMP and SuperGLUE scores, with the latter multiplied by 100 for scale adjustment.

Dataset	Epochs	BoolQ	MNLI	MultiRC	RTE	WSC	MRPC	QQP	BLiMP	S. GLUE	Average
Human 1M	10	0.691	0.442	0.660	0.597	0.635	0.721	0.699	54.38	**0.635**	58.94
	50	0.681	0.425	0.658	0.554	0.673	0.706	0.696	57.66	0.628	60.21
	500	0.698	0.419	0.663	0.547	0.615	0.711	0.688	57.12	0.620	59.56
Best synthetic 1M	50	0.696	0.429	0.667	0.583	0.615	0.730	0.700	**57.82**	0.631	**60.48**
Human 10M	10	0.698	0.449	0.670	0.554	0.654	0.770	0.725	69.38	0.646	66.97
	50	0.694	0.458	0.668	0.576	0.654	0.730	0.745	**71.68**	0.646	**68.15**
Best synthetic 10M	50	0.702	0.509	0.673	0.619	0.654	0.735	0.758	63.76	**0.664**	65.10

Table 2: Results of evaluation of ModernBERT 149M trained on human text corpora (BabyLM) compared to the best model (acc. to average) trained on synthetic data of the same size.

5.2 Results

The evaluation results for models trained on data synthesized by our method are presented in Table 1.

Analyzing model performance as the average across both benchmarks, multi-task pretraining achieved the best results for all model and dataset sizes, as well as for all training durations measured in epochs. The only exception was the 149M-parameter model trained on the 10M corpus with a computation budget of 50 epochs, where training on the vocabulary-constrained corpus yielded the best average score, although it was still outperformed by multi-task pretraining on BLiMP.

The comparison between text generation and vocabulary-constrained corpora does not reveal a clear winner, as both approaches produced very similar results across all tested configurations. Likewise, we did not observe substantial performance differences between the two studied model sizes. However, the comparison of the best average results indicates that slightly better outcomes were achieved with the larger model for both corpus sizes.

Comparing models trained on the 1M corpus for 500 epochs and the 10M corpus for 50 epochs (both exposed to the same total number of tokens), we observe clear benefits from training on more diverse texts rather than repeatedly reusing the same content. BLiMP improves by 7 percentage points, while SuperGLUE increases by about 3 percentage points. Interestingly, the best average results among models trained on 1M-word corpora were obtained with a 100-epoch budget, suggesting no clear benefits from increased training time.

Comparison with pretraining on human-written corpora To compare the effectiveness of training on our synthetic datasets with training on human-written texts, we trained the 149M version of our model on BabyLM corpora (Charpentier et al., 2025) using the same model hyperparameters. We took the 10M-word version of the BabyLM corpus (denoted as Human 10M) and additionally used the first 1M tokens of it as the smaller, 1M-word version (Human 1M). Table 2 reports the evaluation results of models trained on human-written corpora, along with the best-performing synthetic-data models for each corpus size, provided as a reference.

For a corpus of one million words, models trained using Human-1M performed worse than the best model trained on synthetic data, measured as an average of GLUE and BLiMP benchmarks. On larger corpora, the models trained on human data obtained higher BLiMP score, but the model trained on synthetic data was still better for fine-tuning on downstream tasks of GLUE benchmark.

Analysis of generated data Basic statistics and vizualizations of generated datasets are presented in App. C. The multi-task dataset contains a significantly larger number of samples with shorter texts in comparison to other corpora. This is because many classification tasks operate on single sentences rather than the paragraphs or documents typically found in text corpora.

As expected, the vocabulary-controlled dataset has the smallest vocabulary, whose frequency distribution has a significantly shorter tail than those of the other studied datasets. The multi-task and text generation datasets both have fewer occurrences of high-frequency words than the human corpus and higher vocabulary diversity.

The vast majority of tasks have imbalanced label distributions. A few tasks have very long-tail distributions of label frequencies and classes with nearly zero instances. About a quarter of the examples in 1M corpora and only 4.7% of them in 10M one have labels for any sequence prediction/tagging task. This is due to the teacher LLM's failure

Dataset	BLIMP	S. GLUE	Average
Multi-task	56.80	0.634	60.11
only text classification	57.08	0.645	60.79
only pair text class.	56.85	0.639	60.38
only tagging	57.67	0.627	60.20

Table 3: Results of evaluation ModernBERT 149M trained for 10 epochs for different versions of 1M words Multi-task corpus.

to generate a sequence of labels of the expected length[3], hindering the possible gains from these tasks.

Ablation study We performed an ablation study to verify which tasks categories contribute the most to the final results. We performed experiments on 1M Multi-task corpus keeping only labels from a selected task category and training for 10 epochs the larger version of our model.

The results are presented in Table 3 and are slightly higher than those for the basic version of the multi-task corpus containing all the labels. Training only on text classification labels yields the highest improvement. This may be related to the fact that this group of tasks has the highest number of generated labels in our corpus, providing labels for almost all instances and thus making the corresponding task heads well-trained.

6 Summary

This paper introduces a method for pretraining language models entirely on synthetic data generated by a large language model (LLM) using fully automatic pipeline. The teacher model automatically design and generate datasets for diverse NLP tasks, spanning across text classification, tagging, and text generation. The additional training information comming from synthetic labels is exploited during training via the proposed multi-task loss.

Experiments with transformer-based language models on SuperGLUE and BLiMP benchmarks demonstrated that fully synthetic, automatically generated multi-task corpora can serve as an effective substitute for human text in pretraining.

Acknowledgments

This work was supported by the European Research Council (Grant agreement No. 101039303, NG-NLG) and used resources of the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).

[3]The structure decoding algorithm allowed for format specification and limited the generation to lists of valid label names, but it could not control the length of the label list

Limitations

Some concerns related to training LLMs on existing text corpora are related to potential copyright and privacy issues associated with using web-scraped content and learning potential biases expressed in the data. Although the presented method do not use any existing text corpora, it exploits an LLM that was trained on web-scraped data, so the generated synthetic data may have similar issues.

This paper was limited in testing different configurations of trained models and it is highly probable that the training parameters used were not optimal.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. *Preprint*, arXiv:2306.08543.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining

on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, Yuxian Gu, Xin Cheng, Xun Wang, Si-Qing Chen, Li Dong, Wei Lu, Zhifang Sui, Benyou Wang, Wai Lam, and Furu Wei. 2024. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *Preprint*, arXiv:2402.13064.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *Preprint*, arXiv:2402.13116.

A Data generation prompts

The prompts used in the data generation pipeline are presented in Listings 1, 2, 3, 4, and 5.

B List of tasks in multi-task corpora

The list of tasks designed by LLM for the multi-task corpora is provided in Tab. 8. The histograms of labels for text classification, pair text classification and tagging tasks are presented in Fig. 3, 4 and 5, respectively. If the number of different labels for a task was higher than 10, the smallest classes was aggregated to "Other" class to keep the figures readable.

C Additional dataset characteristics

Basic corpora statistics for 1M datasets are presented in Table 4 and in Table 6 for 10M. Additionally, basic label statistics for 1M Multi-task dataset are provided in Table 5 and in Table 7 for 10M.

The histograms of text lengths in studied 1M corpora are presented in Fig. 6. Word frequency distributions in the studied 1M corpora are shown in Fig. 7.

```
    You are a large language model teaching a smaller language model everything you
    know. The smaller model should only cover English, so exclude anything related to
    other languages (translation, coding, language identification, etc.).

Your student's study plan contains several stages that will enable them to learn how
to perform all the tasks that you can perform.

You are currently planning a learning stage involving text classification tasks. These
tasks require a single text as input and provide a single class as output. Note that
this learning stage should not include any other tasks, such as the classification of
pairs of texts or sequences.

Please generate an exhaustive list of text classification tasks, which should provide
sufficient material for creating a versatile language model.
```

Listing 1: Task generation prompt for creating a list of text classification tasks.

```
    Generate an annotation schema of a dataset for {name} task.
Task description: {description}

Since this is a {task_type} task, an annotation scheme is simply a list of all
possible classes. The classification should be fine-grained, but the classes should be
precisely defined so there is no ambiguity in the labeling. It is better to have fewer
classes that are well defined than many classes that are not clearly defined.
```

Listing 2: Prompt for designing annotation schema for different tasks.

```
I want to generate artificial dataset for a machine learning task using an LLM. Here
are the details of the task:

===
TASK INFORMATION
The task is called: {task_name}.
Task description: {task_description}.
===

Could you give me a long list of possible prompts that would make an LLM generate
about 50 test examples (i.e. possible inputs)? The prompts should be clear and
diversified.  You can assume that the LLM is already aware of the information given in
'TASK INFORMATION' section provided above. While you can include a few examples in
some prompts, remember that your task is to create the prompt for the LLM, NOT to
generate the data.
```

Listing 3: Prompt for designing a list of prompts for a given task.

```
You are a data generation assistant. Your role is to create high-quality synthetic
data tailored to the user's specifications. You generate data that is realistic,
diverse, and suitable for tasks such as machine learning training, testing, and
simulation.

Your ONLY functionality is to generate diversified data for the following task:

Task name: {task_name}
Task description: {task_description}
Task type: {task_category}
Tag set: {task_class_list}

For each given prompt, you should respond with a list of 50 examples. {type_dsc}

Do not include explanations unless explicitly asked. Only return raw data or
structured output as specified. While generating examples, take into account the given
user prompts, but remember that producing data that follow the task specification
given above is crucial.
```

Listing 4: System prompt for the generation of training examples.

```
You are a text classifier, for the following task:

Task name: {task_definition['name']}
Task description: {task_definition['description']}
Task type: {task_definition['task_type']}
Tag set:
    {class_dsc}

For each given pair of input sentences provided by the user, classify it into one of
the following categories: {self.list_of_labels}.

For a given input, respond with a JSON object that matches the following schema:
{format_dsc} where label is one of the labels from the tag set.

Don't respond with any explanations, just return the JSON object.
```

Listing 5: Prompt used to construct a classifier for a given task.

Dataset	Samples	Total Chars	Avg Char	Median Char	Max Char	Avg Token	Median Token	Max Token
Text gen.	2529	7049642	2788	2894	9193	570	584	2510
Multi-task	38055	6576964	173	90	6096	36	18	1262
Vocab. c.	1187	5677209	4783	4698	10913	1078	1042	2094
Human	5007	5207559	1040	1046	1,531	254	254	255

Table 4: Basic characteristics of used 1M corpora. Number of samples and text length statistics measured in tokens and characters. Human corpora was provided as a free-text, without division into samples – we treated a training batch as a sample to compute these statistics.

Task	Unique Tasks	Samples With Task	Unique Labels	Total Labels
Text Classification	19	22889	263	313139
Text Pair Classification	12	12012	105	97152
Sequence Prediction	6	9891	78	81650

Table 5: Basic characteristics of task labels generated in 1M Multi-task dataset.

Dataset	Samples	Total Chars	Avg Char	Median Char	Max Char	Avg Token	Median Token	Max Token
Text gen.	25,246	70,492,774	2792	2910	9,193	570	584	2,510
Multi-task	641,324	131,874,012	206	95	7,305	43	19	1,860
Vocab. c.	11,927	56,711,626	4755	4672	12,289	1077	1043	2,338
Human	67,740	54,202,906	800	814	1,563	254	254	255

Table 6: Basic characteristics of used 10M corpora. Number of samples and text length statistics measured in tokens and characters. Human corpora was provided as a free-text, without division into samples – we treated a training batch as a sample to compute these statistics.

Task	Unique Tasks	Samples With Task	Unique Labels	Total Labels
Text Classification	19	185204	263	475454
Text Pair Classification	12	100552	105	185692
Sequence Prediction	6	30394	78	265910

Table 7: Basic characteristics of task labels generated in 10M Multi-task dataset.

Table 8: List of tasks in the generated dataset

Task Type	Task Name	#Classes	Description
text classification	Movie Review Sentiment Analysis	3	Classify movie reviews as positive or negative.
text classification	Product Review Sentiment Analysis	12	Classify product reviews as positive, negative, or neutral.
text classification	Political Speech Sentiment Analysis	3	Classify political speeches as positive, negative, or neutral.
text classification	Email Spam Classification	3	Classify emails as spam or non-spam based on their content.
text classification	Text Message Spam Classification	2	Classify text messages as spam or non-spam based on their content.
text classification	Topic Modeling	25	Classify articles into topics like science, technology, politics, sports, entertainment, etc.
text classification	Product Category Classification	58	Classify products into categories like electronics, clothing, home goods, etc.
text classification	Emotion Classification	5	Classify text as happy, sad, angry, surprised, or fearful.
text classification	Intent Classification	15	Classify text as booking a hotel room, making a reservation, asking for directions, etc.
text classification	Aspect-Sentiment Analysis	10	Identify aspects of a product (e.g., quality, price, design) and classify the sentiment towards each aspect.
text classification	Hate Speech Classification	4	Classify text as hate speech or not.
text classification	Toxic Content Classification	13	Classify text as toxic or not.
text classification	Product Recommendation	5	Classify text as recommending a product or service.
text classification	Question Type Classification	9	Classify questions as fact-based, opinion-based, or open-ended.
text classification	Text Summarization Classification	2	Classify text as a summary or not.
text classification	Fake News Classification	5	Classify news articles as fake or real.
text classification	Medical Condition Classification	17	Classify text as describing a specific medical condition.
text classification	Occupation Classification	12	Classify text as describing a particular occupation.
text classification	Location Classification	60	Classify text as describing a specific location.
text pair classification	Entailment Tasks	4	Determine if one text implies or supports another.
text pair classification	Recognizing Textual Entailment (RTE)	3	Similar to textual entailment but more challenging.
text pair classification	Question Pair Classification	10	Classify question types and answer types.
text pair classification	Text Similarity and Dissimilarity	6	Measure how similar two texts are in terms of meaning.
text pair classification	Contrasting Texts	5	Identify pairs of contrasting statements.
text pair classification	Emotion and Sentiment Analysis	10	Classify emotional tone and determine sentiment polarity.
text pair classification	Coherence and Consistency	5	Evaluate coherence and detect inconsistencies.
text pair classification	Argumentation and Debate	7	Assess argument strength and detect persuasion.
text pair classification	Factuality and Veracity	5	Verify facts and evaluate trustworthiness.
text pair classification	Identity and Intent	16	Identify authors and speakers, and infer intent.
text pair classification	Relationship and Entity Classification	27	Extract relationships and resolve coreferences.
text pair classification	Text Generation and Editing	7	Evaluate grammaticality and fluency.
sequence prediction	Part-of-Speech (POS) Tagging	17	Identify the grammatical category of each word in a sentence.
sequence prediction	Named Entity Recognition (NER)	9	Identify named entities in a sentence.
sequence prediction	Chunking or Phrase Chunking	13	Identify phrases or chunks within a sentence.
sequence prediction	Dependency Parsing	20	Analyze the grammatical structure of a sentence.
sequence prediction	Semantic Role Labeling (SRL)	12	Identify the roles played by entities in a sentence.
sequence prediction	Coreference Resolution	7	Identify pronouns and their antecedents.
text generation	Text Summarization	0	Given a long piece of text, generate a concise summary while preserving essential information.
text generation	Article Generation	0	Write an original article on a given topic, including introductory paragraphs, main content, and conclusion.
text generation	Storytelling	0	Create a short story based on a prompt, including characters, setting, plot, and resolution.
text generation	Dialogue Generation	0	Generate conversations between two or more people on a specific topic or scenario.

Task Type	Task Name	#Classes	Description
text generation	Product Description Writing	0	Craft compelling product descriptions based on product specifications, features, and benefits.
text generation	Social Media Post Generation	0	Write engaging social media posts, including captions and hashtags, for a variety of topics and platforms.
text generation	Email Response Generation	0	Respond to emails with a personalized message, addressing the sender's concerns or questions.
text generation	Chatbot Conversations	0	Engage in natural-sounding conversations with users, providing relevant information and support.
text generation	Poetry Generation	0	Create original poems based on prompts, using various forms and styles.
text generation	News Article Rewriting	0	Rewrite news articles in different tones, styles, or formats while maintaining the same facts.
text generation	Speechwriting	0	Write speeches for various occasions, such as weddings, graduations, or business presentations.
text generation	Book Reviews	0	Generate reviews of books, including summaries, analysis, and opinions.
text generation	Recipe Writing	0	Create recipes with step-by-step instructions, ingredient lists, and nutritional information.
text generation	Travel Itinerary Planning	0	Plan travel itineraries, including suggested routes, activities, and accommodations.
text generation	Mad Libs	0	Fill in missing words in a story or sentence with the correct parts of speech (e.g., noun, verb, adjective).
text generation	Creative Writing Prompts	0	Complete writing prompts that encourage creative thinking and storytelling.
text generation	Transcription	0	Transcribe spoken text into written form, maintaining accuracy and clarity.
text generation	Caption Writing	0	Write captions for images, videos, or memes, conveying the essence of the content.
text generation	Conversation Flow	0	Generate conversation flows for various scenarios, ensuring a logical and coherent discussion.
text generation	Scriptwriting	0	Write scripts for movies, plays, or TV shows, including dialogue and scene descriptions.

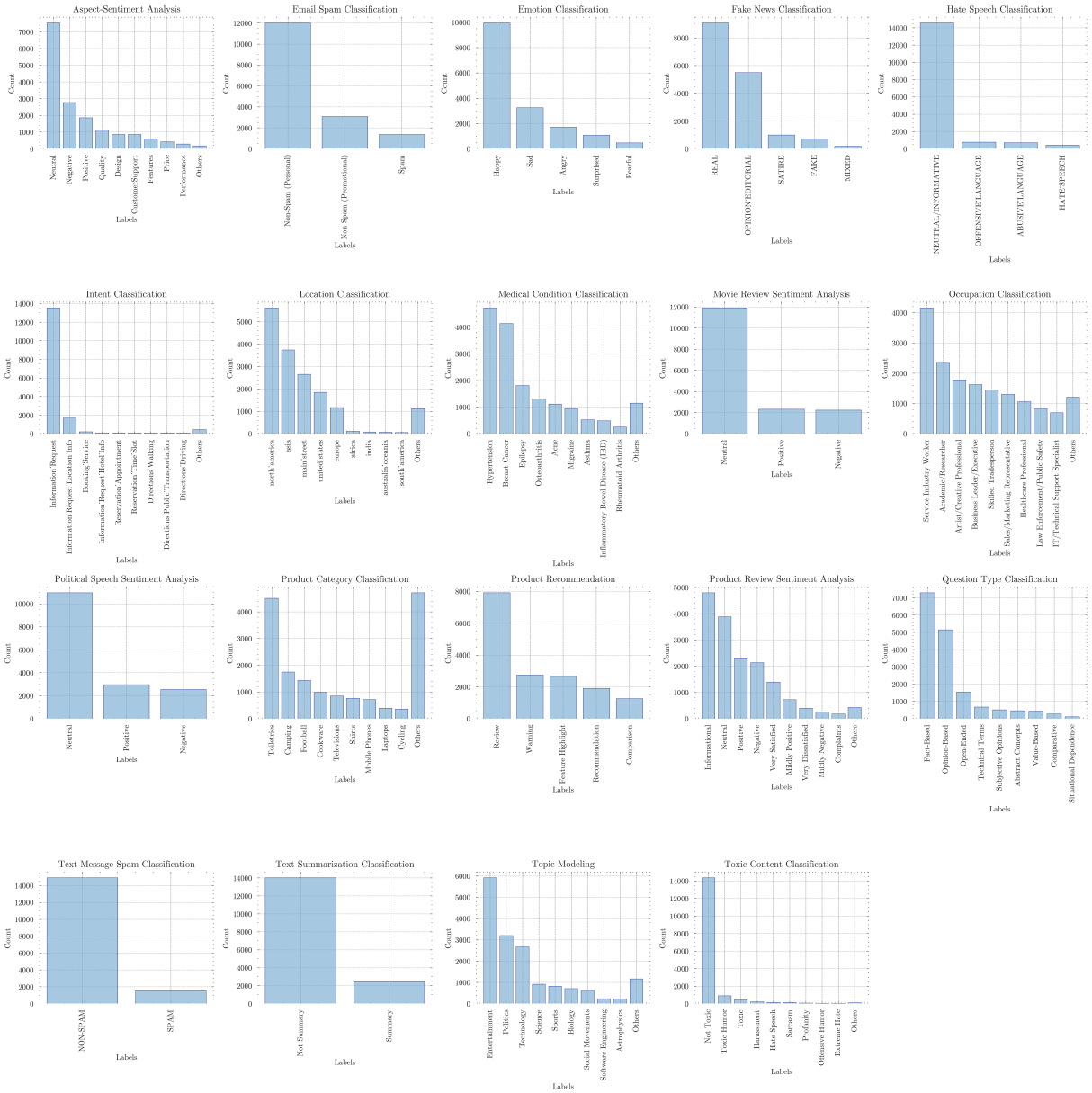


Figure 3: Histograms of labels for text classification tasks in the Multi-task corpus (1M words).

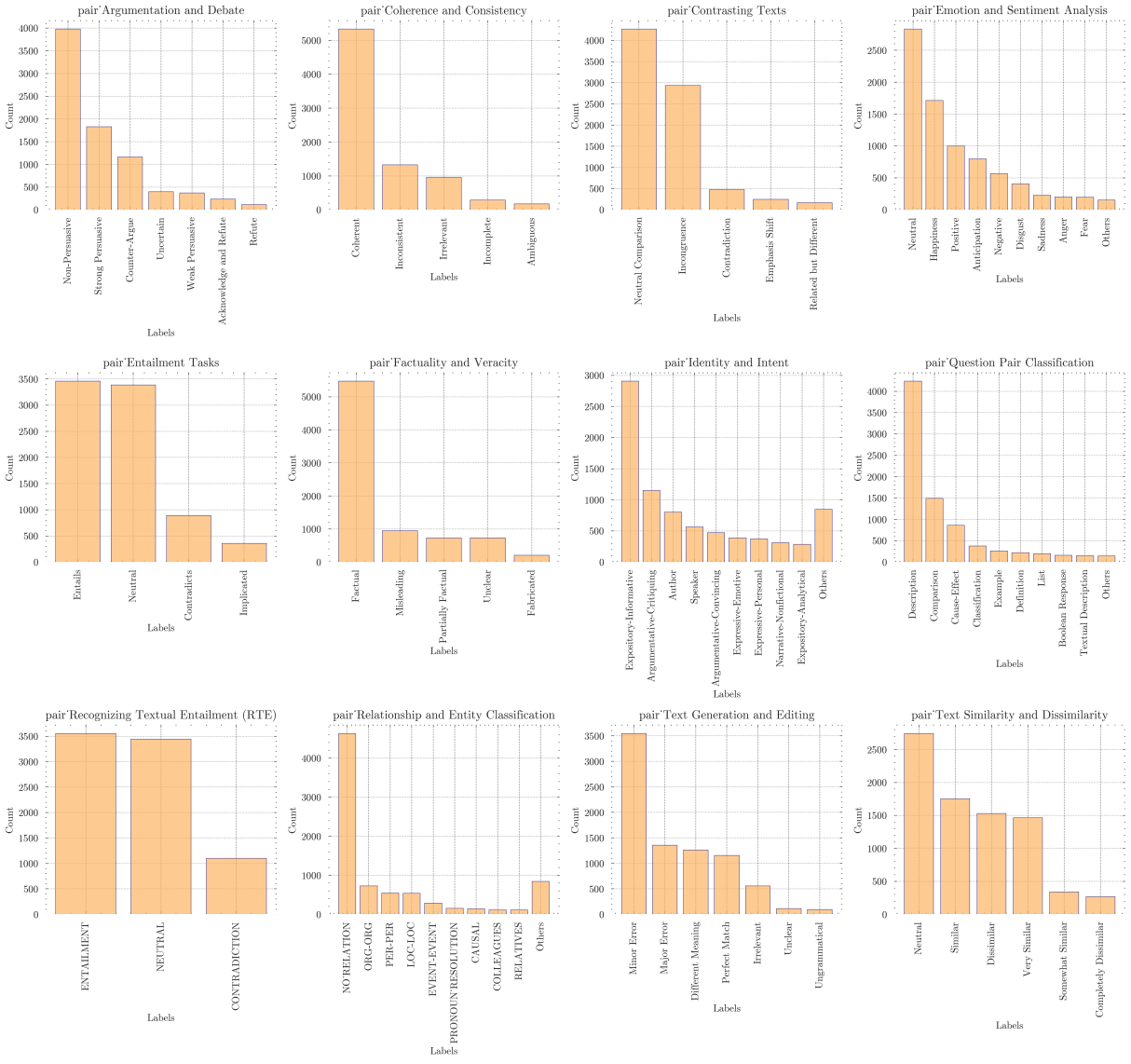


Figure 4: Histograms of labels for pair text classification tasks in the Multi-task corpus (1M words).

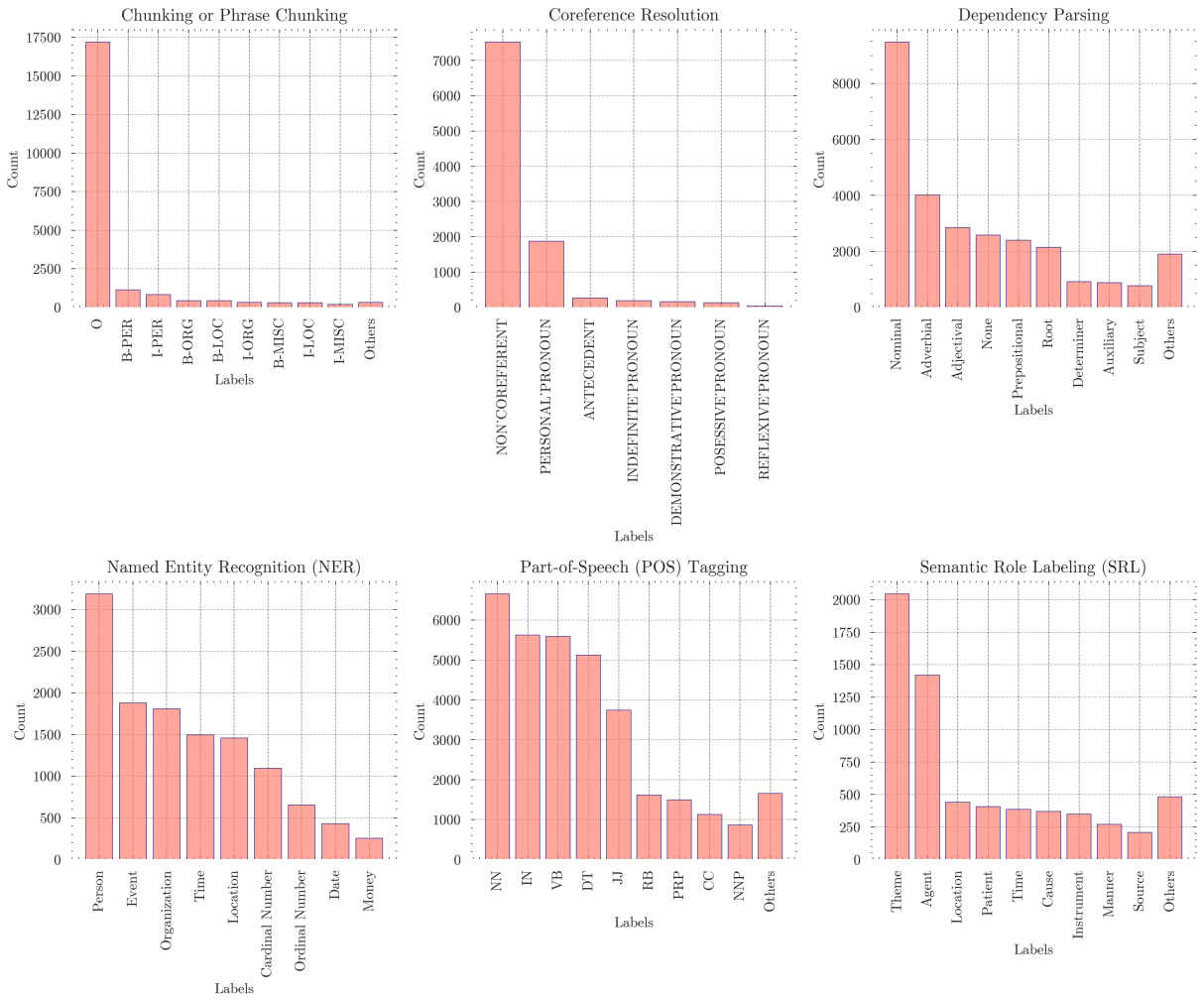


Figure 5: Histograms of labels for tagging tasks in the Multi-task corpus (1M words).

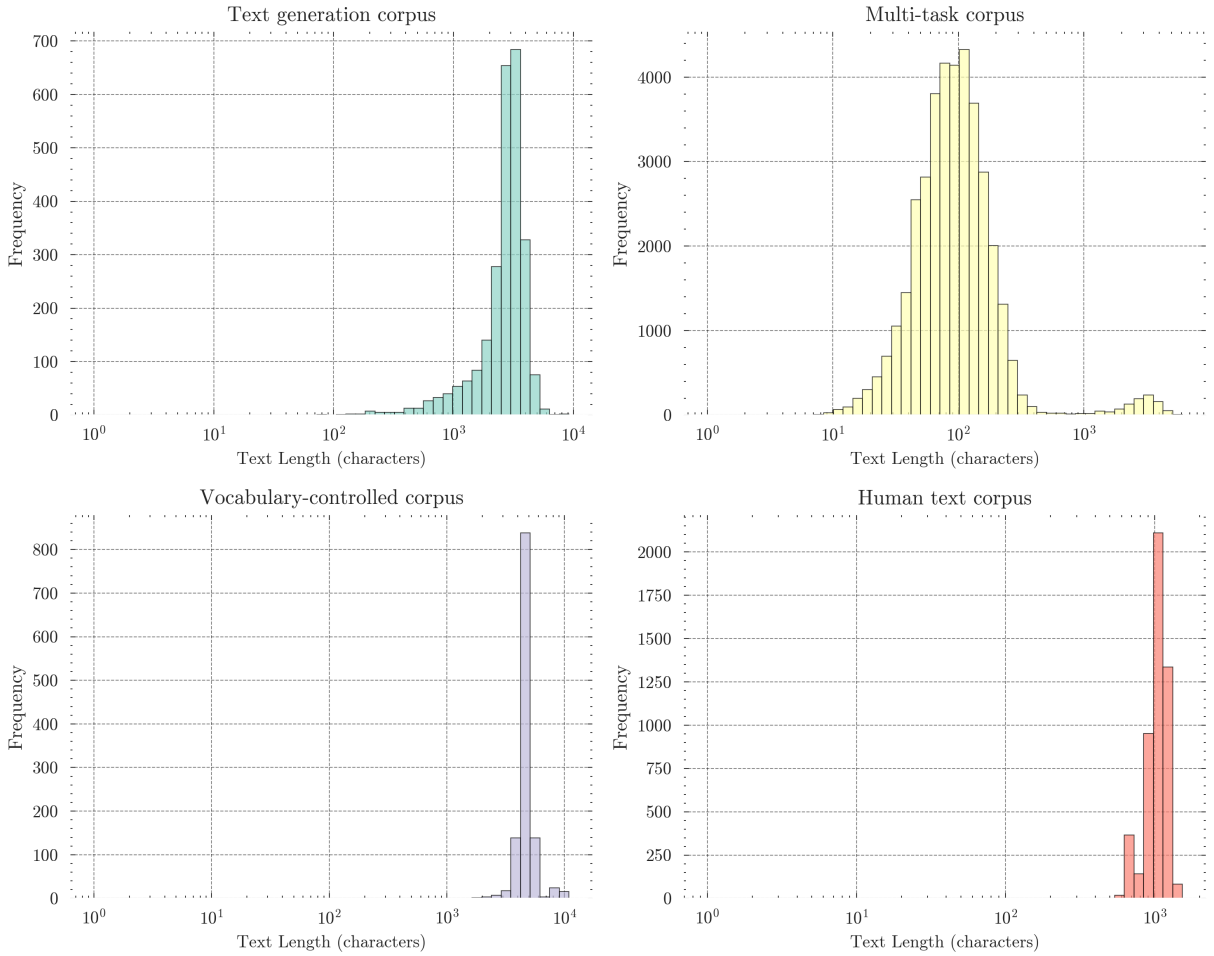


Figure 6: Histograms of text lengths (measured in characters) for different datasets.

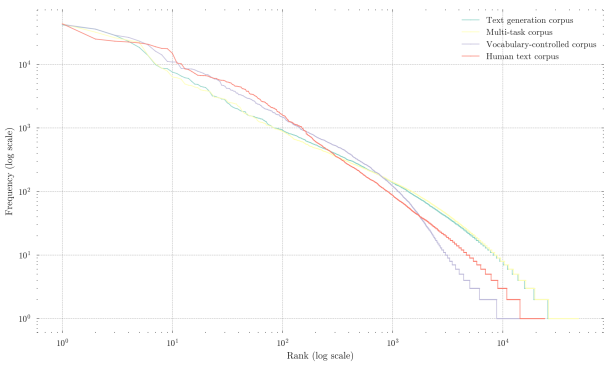


Figure 7: Word frequency in tested corpora.

Active Curriculum Language Modeling over a Hybrid Pre-training Method

Eleni Fysikoudi, Sharid Loáiciga, Asad Sayeed
Department of Philosophy, Linguistics and Theory of Science,
University of Gothenburg
sharid.loaiciga@gu.se, gusfysel@student.gu.se, asad.sayeed@gu.se

Abstract

We apply the Active Curriculum Language Modeling (ACLM) method to the constrained pretraining setting of the 2025 BabyLM Challenge, where models are limited by both data and compute budgets. Using GPT-BERT (Charpentier and Samuel, 2024) as the base architecture, we investigate the impact of surprisal-based example selection for constructing a training curriculum. In addition, we conduct a targeted hyperparameter search over tokenizer size and batch size. Our approach yields stable pretrained models that surpass the official baseline on multiple evaluation tasks, demonstrating ACLM's potential for improving performance and generalization in low-resource pretraining scenarios.

1 Introduction

We present our submission to the BabyLM Challenge 2025[1]. Now in its third edition, the BabyLM Challenge invites participants to investigate how language models can be trained under data constraints that mirror those of human learners. The core shared task includes two text-only tracks, 100M `strict` and 10M `strict small`, that limit the amount of training data to developmentally plausible levels. There is also a multimodal track that broadens the scope to vision and language learning, and this year introduces a new interaction track, where agents learn through dialog with each other.

Our submission targets the `strict small` track only and builds on the Active Curriculum Language Modeling (ACLM) model described in Hong et al. (2023, 2024). The model relies on GPT-BERT (Charpentier and Samuel, 2024) as base architecture combined with active learning and a learning schedule. Although the approach did not obtain competitive performance in previous years, the shared task this year introduces new compute limitations: models may train for no more than 10 epochs, i.e, may not be exposed to more than 100M words in total during training (Charpentier et al., 2025). In this context, we test whether the ACLM approach is more effective, as the active learning criterion, in which the model self-selects the sentences it is most confused about, should normally improve learning efficiency.

Our results demonstrate that the ACLM method produces stable pretrained models which outperform a vanilla GPT-BERT baseline on certain tasks. In addition, we conducted a targeted hyperparameter search, focusing primarily on the tokenizer size and batch size, to optimize performance and investigate how these parameters affect the ACLM algorithm.

ACLM is intended to be a wrapper around other, usually Transformer-based, language modeling paradigms. In submitting an ACLM-based model to the 2025 BabyLM task, we examine whether ACLM makes a difference relative to the baseline set by the most successful language modeling approach from the 2024 task. This task report provides our final results before the task deadline, which show that it continues to be fruitful to explore the potential for using dynamic approaches to selecting training instance order despite advances in the underlying LLM technology.

2 Related work

GPT-BERT (Charpentier and Samuel, 2024), the winning submission of the BabyLM Challenge 2024, combines the strengths of autoregressive (GPT-style) and masked (BERT-style) language modeling in a single architecture that can switch between the two training modes without extra parameters. Charpentier and Samuel report consistent better performance than both masked-only and causal-only models when training on the 2024 BabyLM

[1] https://babylm.github.io/

Proceedings of the First BabyLM Workshop, pages 500–507

data.

GPT-BERT aligns both masked and causal language modeling through masked next-token prediction (MNTP), a variant from traditional masked language modeling (MLM) where predicting a masked token at position $k + 1$ is predicted at position k. This means that there are two modes or training objectives but a single LTG-BERT architecture (Samuel et al., 2023) without additional parameters. The data is duplicated to ensure that both objectives are exposed to all the data and the model is trained using cross-entropy loss. Additional improvements to the base architecture include: i) attention output gating, where attention is modulated through GEGLU activation function (Shazeer, 2020); ii) layer weighting from ELC-BERT (Charpentier and Samuel, 2023), where each layer learns linear combinations of outputs from previous layers, as opposed to treating all layers equally; iii) batch-size scheduling, starting with smaller batches and lineraly increasing up to 4M tokens to improve efficiency; and iv) mask scheduling, gradually reducing the masking rate from 30% to 15% (the standard) during training.

3 Method

Active Curriculum Language Modeling (ACLM) is a means of dynamically controlling the training schedule introduced by Hong et al. (2023) and developed further in Hong et al. (2024). ACLM is inspired by more "classic" ideas in machine learning, such as active learning and curriculum learning (Jafarpour et al., 2021). Active learning was developed for classification problems, where the artificial learner was designed in such a way as to be able to identify the unlabeled data it was least confident about, allowing human annotators to work on a smaller set. Curriculum learning involves a schedule of training data set in advance. As language modeling is not a learning problem with a fixed set of categorical classes, ACLM adapts the active learning paradigm instead to automatically select the token sequences that share the same uncertainty characteristics as previously seen training instances. Human intervention between training epochs, as in "classic" active learning, is thereby eliminated, leading to a curriculum that is updated dynamically over the course of the training process, reflecting an intuition that language acquisition is an interactive and dynamic process (Masek et al., 2021) in which children are active participants in driving the organization of the stimulus (Saylor and Ganea, 2018).

Figure 1 depicts the ACLM architecture, with further elaboration in algorithms 1 and 2. In an initialization phase, a randomly-selected subset of training instances is taken from the overall training pool. These are used to train an initial model. At the same time, all training instances in the corpus (which, in this work, all have equal token length) are transformed into vectors of surprisal (negative log-probability given the context) values for each token. That is, $(w_1, w_2, \ldots, w_n)$ is converted to $(s_1, s_2, \ldots, s_n)$ where $s_n = -\log P(w_n|C)$ where C is the context used by the language model, which varies depending on the specific language model; this can normally be computed from the model's cross-entropy loss at each token. These vectors are sorted into a "surprisal space" which can be queried by k-Nearest Neighbours algorithms.

The transition to future epochs involves selecting the already-trained instance q that exhibits the highest surprisal given the context and the current state of the model. The surprisal space is queried to present a subset of unseen instances that are most similar to q in terms of surprisal, and these become the training subset for the next ACLM iteration. That is, the next subset is not chosen for its direct "semantic" similarity to q, but rather in terms the similarity of their patterns of uncertainty (represented as sequences of surprisal values) to q's pattern of uncertainty. The underlying intuition is that the learner seeks out instances that are *similarly uncertain*, rather than instances that are merely only similar to q.

In our implementation [2], the initial surprisal space is bootstrapped with a simple trigram-based token probability model. Later ACLM iterations update the surprisal vectors based on their current state, producing an interatively dynamic curriculum. The ACLM process is intended to be wrapped around a specific language modeling paradigm. This year, we have wrapped ACLM around GPT-BERT. While in an ideal world, this should be a fully modular process, in practice, LLM implementations differ in their input intake and their provision of output values, requiring nontrivial adaptation effort. The most important modification from the original GPT-BERT result was that in the 1:3 and 1:7 ratio settings, the dataset was no longer

[2] https://github.com/elenifysikoudi/gpt_bert_ACLM

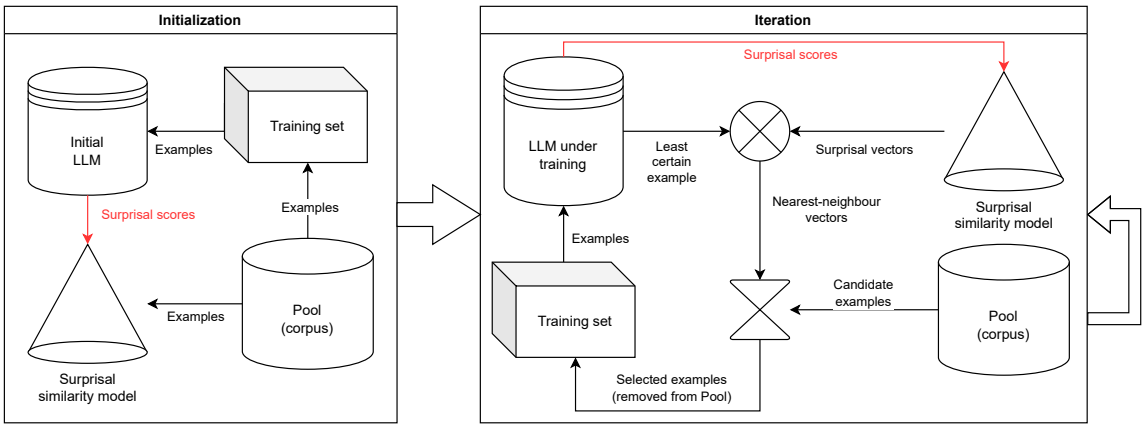


Figure 1: The architecture of our ACLM method from last year's submission, described in Hong et al. (2024).

Algorithm 1 Initialization phase of the ACLM process (after Hong et al., 2024).

```
Model ← new(GPT-BERT)
ActiveSet ← select_random(Pool, n_initial)
train(Model, ActiveSet, n_epochs)
SurprisalSet ← []
for all instances i in Pool do
    surprisals ← Model.surprisals(i)
    SurprisalSet.append(surprisals)
end for
```

Algorithm 2 Iterations of the ACLM process. The kNN procedure also removes the instances from the Pool (after Hong et al., 2024).

```
for iter ← 0 to n_iterations do
    max_surprised ← TrainingSet[0]
    for all instances i in TrainingSet do
        orig_surprisal ←
          Model.surprisals(max_surprised)
        new_surprisal ← Model.surprisals(i)
        if orig_surprisal < new_surprisal then
            max_surprised ← i
        end if
    end for
    ActiveSet.update(SurprisalSet.kNN(
            max_surprised, k, Pool))
    train(Model, ActiveSet, n_epochs)
    SurprisalSet ← []
    for all instances i in Pool do
        surprisals ← Model.surprisals(i)
        SurprisalSet.append(surprisals)
    end for
end for
```

jointly distributed across GPUs; instead, each GPU processed the dataset independently. This was due to differences between the implementations of the underlying machine learning architecture on AMD chips (used by the original GPT-BERT submission) and the NVidia chips to which we had access. It is highly plausible that this implementation difference influenced our results. Furthermore, while we use surprisal as a cognitively-motivated statistic (Fazekas et al., 2020), it is possible to replace this statistic with other values derivable from a language model.

4 Results

4.1 Shared task evaluation

We present the results obtained with the official shared task evaluation scripts in Table 1 for the fine-tuning setting and Table 2 for the zero-shot.

We introduce a constrained GPT-BERT baseline, where the key difference from the official setup is that the sequence length is fixed at 128 throughout training rather than being gradually increased. ACLM outperforms this baseline across all categories, with the notable exceptions of BLiMP and the WUG past-tense correlation task. Furthermore, our two final submissions to the official Hugging Face leaderboard (highlighted rows in Tables 1 and 2) also tended to outperform several of the provided baselines. The highest-ranking model, *gpt_bert_ACLM_mixed_4k*, was trained with a 1:1 (50:50) causal-to-masked objective ratio using a 4K token BPE tokenizer and a batch size of 64. You can also find the rest of the hyperparameters in Section A. We evaluated the model on both the causal and MNTP backends, with the causal backend achieving a substantially higher overall text

score of 39.1. This performance surpasses both the GPT-2 baseline and the GPT-BERT masked-focus baseline, and falls just below the next strongest baseline, which scores 39.2.

4.2 Hyperparameter experiments and analysis

Given the limitations of our computing infrastructure, we experimented with varying causal-to-masked objective ratios, batch sizes, and vocabulary/tokenizer sizes.

We observe that a 1:3 (75:25) causal-to-masked objective ratio generally yields the highest fine-tuning scores on some individual tasks, depending on the specific hyperparameters. For example, the 1:3 model with a 4k tokenizer achieves the best scores on MNLI, MRPC, and RTE compared to other models. Nevertheless, the 1:1 ratio still dominates in terms of the overall GLUE average.

Regarding the Age of Acquisition (AoA) task, due to time constraints we were only able to test a limited set of models, which mostly scored in the range of -0.07 to 0. An exception is the *gpt_bert_ACLM_mixed_4k* model, which achieved a score of 10.04, as reported on the leaderboard.

On the zero-shot tasks, our models perform comparably to or better than the leaderboard baselines and our constrained GPT-BERT models on EWOK, Entity Tracking, COMPS, and Reading. A particularly noteworthy result is the WUG adjective nominalization, where scores range from 61 up to 79. This relatively high correlation highlights the extent to which the model's generalization behavior aligns with human-like patterns.

Overall, these findings suggest that ACLM can be a beneficial pretraining method even under constrained training regimes. GPT-BERT models wrapped around the ACLM framework with mixed objectives can approach——or in some cases surpass——established baselines, while displaying promising signs of human-like generalization.

5 Discussion

The results indicate that smaller batch sizes are more effective for fine-tuning. Similarly, smaller vocabularies tend to yield better performance, with a size of 4k producing strong results on GLUE and 6k performing well on reading times and entity tracking. These findings are consistent with Oh and Schuler (2025), who report that vocabularies in the range of 4k to 8k possess the greatest predictive power with respect to surprisal. In a similar vein, shorter sequence lengths prove advantageous under constrained settings: a length of 128 tokens is sufficient, while increasing to 512 tokens yields only marginal or negligible improvements, as reflected in our results tables.

Regarding the balance between causal and masked objectives, Charpentier and Samuel (2024) report that their best-performing configurations were obtained in multi-GPU training settings with causal-to-masked objective ratios of 1:3 (75:25), 1:7 (87:13), and 15:16 (93:7). In contrast, our experiments indicate that the ACLM method performs best at a 1:1 (50:50) ratio in a 2-GPU setting. Moreover, we observe that evaluation under causal backends tends to yield superior results overall.

Beyond the final outcomes, an examination of intermediate checkpoints highlights several notable training dynamics. Models typically exhibit rapid initial convergence, often between 20M and 50M tokens and in some cases even earlier. This accelerated learning, however, is frequently followed by performance plateaus, suggesting the need for strategies to sustain progress, such as higher weight decay or stronger regularization.

BLiMP scores demonstrate a gradual and consistent increase during training—for example, our best *gpt_bert_ACLM_mixed_4k* batch size 64 model improves from 49.3 to 56.5. In contrast, entity tracking exhibits pronounced instability: in certain settings (e.g., 1:7 ratio with a 6k vocabulary), performance rises to 41.8 at 20M tokens before collapsing to 13.4 by the end of training. Models trained with a 50:50 ratio, on their part, are more stable, typically experiencing only minor decreases of around 5%. Interestingly, the 1:1 ratio models appear more resistant to such degradation than the 1:3 and 1:7 configurations, which may be attributable to their longer effective training spans combined with smaller increments of information per update.

Taken together, these findings suggest that smaller batch sizes and more frequent gradient updates contribute to both stability and generalization in resource-constrained training environments. It is also worth noting that our primary focus was on pretraining; consequently, we did not conduct a systematic hyperparameter search during fine-tuning. Such an exploration could potentially have yielded stronger downstream results.

Ratio	Method	Seq Length	Tokenizer	Batch Size	BOOLQ	MNLI	MRPC	MultiRC	QQP	RTE	WSC	GLUE Avg
1:1	ACLM Max	128	8192	256	64.6	40.8	70.6	64.7	71.0	56.8	61.5	61.5
1:1	ACLM Max	128	6k	256	65.4	39.3	70.1	66.4	69.6	58.3	61.5	61.5
1:1	ACLM Max	128	8192	64	66.4	35.8	70.1	64.5	70.2	62.6	61.5	61.6
1:1	ACLM Max	128	4k	64	65.4	39.1	71.1	65.7	70.9	61.9	63.4	62.5
1:1	ACLM Max	128-512	8192	256	65.3	40.9	70.1	63.7	69.6	61.2	63.5	62.0
1:1	ACLM Max	128	4k	256	65.2	39.2	71.1	64.8	70.6	61.9	63.5	62.3
1:1	GPT-BERT	128	8192	256	66.4	39.6	70.1	65.1	70.1	59.7	63.5	62.1
1:1	ACLM Min	128	4k	64	64.9	39.6	71.1	59.1	70.0	60.5	63.5	61.2
1:1	ACLM Min	128	6k	256	64.1	38.5	70.5	62.4	71.3	58.2	63.4	61.2
1:3	ACLM Max	128	8192	256	65.0	39.6	69.6	65.0	70.2	58.3	63.5	61.6
1:3	ACLM Max	128	4k	256	65.8	41.9	72.5	59.4	68.9	64.7	61.5	62.1
1:3	ACLM Max	128	6k	256	66.7	40.1	70.1	66.2	70.9	59.0	61.5	62.1
1:3	ACLM Max	128	8192	64	65.9	34.8	71.6	62.1	70.4	54.7	63.5	60.4
1:3	ACLM Max	128	4k	64	64.9	37.4	69.1	63.6	70.1	60.4	61.5	61.0
1:3	ACLM Max	128	6k	64	64.3	38.4	71.6	63.7	70.9	57.6	61.5	61.1
1:3	ACLM Max	128-512	8192	256	65.1	38.3	71.6	65.7	70.1	58.3	63.5	61.8
1:3	ACLM Max	128-512	6k	256	66.4	40.3	71.6	65.2	71.3	58.3	63.5	62.4
1:3	GPT-BERT	128	8192	256	65.7	39.3	69.6	60.4	70.1	56.1	63.5	60.7
1:7	ACLM Max	128	8192	256	65.3	36.2	70.6	65.1	70.7	59.0	63.5	61.5
1:7	ACLM Max	128	6k	256	65.7	41.4	69.6	64.6	70.1	57.6	61.5	61.5
1:7	ACLM Max	128	8192	64	64.2	35.1	71.6	65.2	71.0	58.3	61.5	61.0
1:7	ACLM Max	128	4k	64	64.5	37.9	69.1	61.2	69.3	54.0	61.5	59.6
1:7	ACLM Max	128	6k	64	64.3	39.1	71.6	58.8	70.6	56.1	61.5	60.3
1:7	GPT-BERT	128	8192	256	65.9	37.4	71.1	62.0	69.3	58.3	65.4	61.3

Table 1: GLUE results for finetuned 100M models. Scores are reported as the average percentage across tasks. "Ratio" stands for masked-to-causal ratio of the base GPT-BERT system. ACLM Max, ACLM Min in the method indicate the heuristic criteria of maximum or minimum surprisal and GPT-BERT indicate our replication and baselines. Systems submitted to the official leaderboard are highlighted in gray.

6 Conclusions

In the context of shared tasks with constrained resources, it is tempting to just "hill-climb"; that is, to adopt the best-seeming approaches from the previous year and to dismiss underperforming or even "failed" approaches. However, BabyLM is also an exploration of how statistical methods, which normally have very demanding data requirements, can be made to approximate human behaviour even in a "stimulus-poor" environment. The implicit reasoning is that, if humans can do it, machines should somehow be able to do it too. Technical fixes that are not *directly* motivated by a cognitively-motivated theory of acquisition will likely always be the lowest-hanging fruit in terms of extracting performance gains in the evaluation metrics—until they eventually run out of "steam".

By extending the ACLM route of BabyLM entries from previous years, this work contributes to the parallel exploration of cognitively-motivated solution spaces to the problem of simulating stimulus-poor language acquisition *in silico*, *given* technical improvements in the artificial language modeling "substrate". This year, we held the overall conditions of the ACLM process to assumptions similar to those made in the implementation of the 2024 ACLM-based BabyLM entry, and we found that there is still potential value in exploring dynamic curricula.

In future work, we recommend branching out from these assumptions. For example, we believe that there is value to be had from exploring criteria other than surprisal, such as variants of outright semantic similarity or entropy reduction (Hale, 2016)—or even linear combinations thereof. Future implementations may also consider other ways of encoding the similar space, such as through clustering the vectors prior to measuring similarity and choosing the nearest neighbours of the nearest centroid, rather than simply the "raw" k-Nearest Neighbours.

Limitations

Even under the constraints of this year's BabyLM challenge, we still face limitations on exploring the entire hyperparameter space, so it is possible that there is a superior combination of hyperparameters that we never got close to. We have some reason to believe that architectural differences between the AMD-based environment of the original GPT-BERT authors and our NVidia-based environment may have an effect on results despite the layers of

Ratio	Method	Seq Length	Tokenizer	Batch Size	BLIMP	Supplement	EWOK	COMPS	Entity Tracking	Reading	WUG	Eval Method
1:1	ACLM Max	128	8192	256	52.8	51.4	49.9	50.1	16.5	7.84/3.96	73/−3	MNTP
1:1	ACLM Max	128	4k	256	55.1	52.4	50.1	50.3	31.6	4.72/3.06	74/−9	MNTP
1:1	ACLM Max	128	8192	64	56.5	50.0	49.8	50.2	16.9	8.25/3.67	70/−26	MNTP
1:1	ACLM Max	128	4k	64	56.5	55.3	49.6	49.9	25.5	4.56/2.54	64/12	MNTP
1:1	ACLM Max	128	4k	64	56.1	53.5	49.4	50.3	32.9	4.66/2.43	70/2	Causal
1:1	ACLM Max	128-512	8192	256	54.3	53.2	49.9	50.5	29.8	8.17/4.17	74/2	MNTP
1:1	ACLM Max	128	6k	256	55.3	50.6	49.9	50.0	33.5	5.45/2.13	79/−2	Causal
1:1	ACLM Max	128	6k	256	54.6	50.0	50.0	49.9	36.0	4.95/1.96	79/4	MNTP
1:1	GPT-BERT	128	8192	256	54.2	52.2	50.0	49.7	16.0	8.08/4.30	73/−5	MNTP
1:1	ACLM Min	128	4k	64	54.3	55.6	49.9	50.7	40.9	4.34/2.71	56/0	MNTP
1:1	ACLM Min	128	4k	64	53.8	51	49.7	50.3	40.9	4.35/2.81	62/−3	Causal
1:1	ACLM Min	128	6k	256	55.3	49.6	49.9	49.9	35.3	5.47/2.36	72/−11	MNTP
1:1	ACLM Min	128	6k	256	57.3	49.8	49.8	50.0	36.7	5.59/2.24	74/-16	Causal
1:3	ACLM Max	128	8192	256	51.6	50.1	50.0	49.9	13.2	8.28/4.15	70/−4	MNTP
1:3	ACLM Max	128	4k	256	51.2	50.8	49.9	49.9	35.1	5.24/3.15	61/−13	MNTP
1:3	ACLM Max	128	6k	256	54.2	53.5	49.8	50.2	19.9	5.33/2.17	75/−6	MNTP
1:3	ACLM Max	128	8192	64	54.6	50.7	49.8	50.2	11.2	8.34/4.15	61/−10	MNTP
1:3	ACLM Max	128	4k	64	55.2	54.8	49.6	50.5	12.2	5.30/3.07	75/7	MNTP
1:3	ACLM Max	128	6k	64	53.2	52.6	50.1	50.2	12.8	5.73/2.53	65/−3	MNTP
1:3	ACLM Max	128-512	8192	256	54.8	51.9	49.9	50.3	13.5	7.50/4.08	65/−7	MNTP
1:3	ACLM Max	128-512	6k	256	54.3	52.0	50.2	50.3	21.3	5.92/2.34	73/0	MNTP
1:3	ACLM Max	128	4k	256	51.9	50.1	50.1	49.9	35.1	5.40/3.21	70/−8	Causal
1:3	ACLM Max	128	4k	256	51.2	50.8	49.9	49.9	35.1	5.24/3.15	61/−13	MNTP
1:3	ACLM Max	128	6k	256	54.5	53.7	50.1	50.0	25.7	5.99/2.39	73/−1	Causal
1:3	ACLM Max	128	6k	256	54.3	52.0	50.2	49.9	21.3	5.92/2.34	73/0	MNTP
1:3	GPT-BERT	128	8192	256	55.3	51.7	50.1	50.5	12.7	7.60/3.88	74/−4	MNTP
1:7	ACLM Max	128	8192	256	54.2	54.4	50.0	50.0	11.8	7.58/4.02	68/11	MNTP
1:7	ACLM Max	128	6k	256	53.0	53.4	50.0	49.8	13.4	5.32/2.19	60/−10	MNTP
1:7	ACLM Max	128	8192	64	53.7	51.2	49.6	50.6	14.0	7.82/3.94	62/5	MNTP
1:7	ACLM Max	128	4k	64	54.5	54.0	49.8	50.4	11.6	4.24/2.83	64/3	MNTP
1:7	ACLM Max	128	6k	64	54.4	54.2	49.9	49.8	12.7	5.88/2.16	66/−3	MNTP
1:7	GPT-BERT	128	8192	256	54.8	51.0	50.2	50.4	12.6	7.52/3.78	71/−5	MNTP

Table 2: Zero-shot results for 100M-parameter models. Scores are reported as average percentages across tasks. "Reading" refers to EyeTrack and SelfPaced, while "WUG" denotes adjective/past. "Ratio" indicates the masked-to-causal ratio of the base GPT-BERT system. AoA is omitted due to time constraints and was only computed for selected models. ACLM Max, ACLM Min in the method indicate the heuristic criteria of maximum or minimum surprisal and GPT-BERT indicate our replication and baselines Systems submitted to the official leaderboard are highlighted in gray.

Python software abstraction separating the code from the hardware.

Acknowledgments

The work reported in this paper has been supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg. Additional funding for this work was provided by the Gothenburg Research Initiative for Politically Emergent Systems (GRIPES) supported by the Marianne and Marcus Wallenberg Foundation grant 2019.0214.

The computations and data storage were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

The authors gratefully acknowledge the support provided by the developers of the GPT-BERT system and the organizers of BabyLM, particularly for their assistance with technical issues and specific inquiries.

References

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Judit Fazekas, Andrew Jessop, Julian Pine, and Caroline Rowland. 2020. Do children learn from their prediction mistakes? a registered report evaluating error-based theories of language acquisition. *Royal Society Open Science*, 7(11):180877.

John Hale. 2016. Information-theoretical complexity metrics. *Language and Linguistics Compass*, 10(9):397–412.

Xudong Hong, Sharid Loáiciga, and Asad Sayeed. 2023. A surprisal oracle for active curriculum language modeling. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 259–268, Singapore. Association for Computational Linguistics.

Xudong Hong, Sharid Loáiciga, and Asad Sayeed. 2024. A surprisal oracle for when every layer counts. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 237–243, Miami, FL, USA. Association for Computational Linguistics.

Borna Jafarpour, Dawn Sepehr, and Nick Pogrebnyakov. 2021. Active curriculum learning. In *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*, pages 40–45, Online. Association for Computational Linguistics.

Lillian R. Masek, Brianna T.M. McMillan, Sarah J. Paterson, Catherine S. Tamis-LeMonda, Roberta Michnick Golinkoff, and Kathy Hirsh-Pasek. 2021. Where language meets attention: How contingent interactions promote learning. *Developmental Review*, 60:100961.

Byung-Doh Oh and William Schuler. 2025. The impact of token granularity on the predictive power of language model surprisal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4150–4162, Vienna, Austria. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Megan Saylor and Patricia Ganea. 2018. *Active Learning from Infancy to Childhood: Social Motivation, Cognition, and Linguistic Mechanisms*.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

A Appendix

Hyperparameter	Submitted model
Number of parameters	31M
Number of layers	12
Hidden size is	384
FF intermediate size	1280
Vocabulary size	4 000
Attention heads	6
Hidden dropout	0.1
Attention dropout	0.1
Training steps	149
Batch size	64
Sequence length	128
Warmup ratio	1.6%
Initial learning rate	0.0141
Final learning rate	0.000141
Learning rate scheduler	cosine
Weight decay	0.1
Optimizer	LAMB
LAMB ϵ	1e-8
LAMB β_1	0.9
LAMB β_2	0.98
Gradient clipping	2.0
Gradient accumulation	16-23

Table 3: Pre-training hyperparameters for the highest scoring GPT-BERT ACLM model trained on the STRICT-SMALL track.

Linguistic Units as Tokens: Intrinsic and Extrinsic Evaluation with BabyLM

Achille Fusco[1,2] **Maria Letizia Piccini Bianchessi**[2]
Tommaso Sgrizzi[2] **Asya Zanollo**[2] **Cristiano Chesi**[2]
[1]University of Florence, Via della Pergola 60, Florence, Italy
[2]IUSS Pavia, Piazza della Vittoria 15, Pavia, Italy
achille.fusco@unifi.it letizia.piccinibianchessi@iusspavia.it
tommaso.sgrizzi@iusspavia.it asya.zanollo@iusspavia.it
cristiano.chesi@iusspavia.it

Abstract

Tokenization is often treated as a preprocessing step, yet in data-limited settings it directly shapes what a model can learn. We compare four segmentation strategies in the BabyLM Challenge: merge-based BPE, morphology-aware, split-based MorPiece and ParadigmFinder, and syllable-based SylliTok. Evaluation combines two perspectives. First, an intrinsic test on the SIGMORPHON 2022 segmentation benchmark, adapted to English, measures how closely each tokenizer aligns with morpheme boundaries. Second, extrinsic tests train GPT-2 on the 10M BabyLM corpus and evaluate it on the 2025 benchmark. No single tokenizer dominates. BPE remains strong on syntax-heavy tasks, ParadigmFinder excels in semantic composition and age-of-acquisition alignment, and SylliTok shows advantages in discourse tracking. Morphology-aware tokenizers achieve the best intrinsic segmentation scores, and these gains translate into more robust generalisation in comprehension tasks. These results highlight tokenization as a core modeling decision, with direct consequences for compression, morphology, and the path to human-like learning.

1 Introduction

The BabyLM Challenge (Warstadt et al., 2023a,b) was designed to evaluate how language models acquire linguistic competence under data conditions that approximate human language learning. By restricting training to corpora of 10M or 100M tokens, the benchmark provides a testbed for exploring which modeling choices enable robust acquisition from limited input. While most submissions have focused on architecture and training objectives, a less visible but equally fundamental choice concerns the unit of tokenization. The segmentation of raw text into model input units determines not only how words are represented, but also what kinds of generalisations the model is in principle able to make.

Standard approaches such as byte pair encoding (BPE; Gage 1994; Sennrich et al. 2016) treat tokenization as a purely statistical compression problem, merging frequent character pairs without regard for linguistic structure. Recent work, however, has argued that tokenization should be viewed as an integral part of the modeling effort (Goldman et al., 2024; Oh and Schuler, 2025), shaping both the inductive biases of the system and its ability to align with humanlike generalisation. In particular, morphology has long been seen as a critical domain for testing theories of language acquisition (Goldsmith, 2001; Xu et al., 2018), and offers a natural arena for designing tokenizers that attempt to capture linguistically meaningful units.

In this paper, we ask how different linguistically oriented tokenizers affect learning in the BabyLM setting. We consider four segmentation strategies, ranging from merge-based (BPE) to split-based (MorPiece, ParadigmFinder) and syllable-based (SylliTok). To evaluate them, we combine two complementary perspectives: an *intrinsic* assessment using the SIGMORPHON 2022 morphological segmentation benchmark, and an *extrinsic* evaluation using the BabyLM 2025 test suite. This dual approach allows us to measure both how well each tokenizer approximates humanlike segmentation and how these choices influence downstream learning and generalisation.

Our results show that no single tokenizer dominates across tasks. Instead, each segmentation strategy introduces its own strengths and weaknesses: morphology-aware tokenizers excel in capturing systematic structure and supporting semantic generalisation, syllable-based segmentation contributes to discourse sensitivity, and frequency-driven BPE remains a strong all-around baseline. Taken together, these findings highlight tokenization as a substantive modeling decision, with implications for compression, morphological generalisation, and

Proceedings of the First BabyLM Workshop, pages 508–519

the design of cognitively plausible learning systems.

2 Motivation

Tokenization is a crucial step in natural language processing, especially in the development and training of language models. It determines the basic units with which models will operate, ultimately shaping their ability to generalize, compress, and understand linguistic structure. While often treated as a technical detail, tokenization in fact sits at the intersection of practical engineering choices, information-theoretic principles, and linguistic theory. In this section, we articulate three perspectives on tokenization: as a modeling choice, as a compression strategy, and as a proxy for morphological segmentation.

2.1 Tokenization as Modeling and Compression

Tokenization is often viewed as a simple preprocessing step—particularly in languages without explicit word boundaries (e.g., Chinese, Japanese, Thai)—but in practice it defines the basic units on which a model learns. Decisions about how to segment text into words or subwords affect the handling of out-of-vocabulary items, the effective sequence length, and the allocation of parameters in the embedding layer. This introduces asymmetries across languages: scripts, morphology and resource availability lead to different degrees of token fragmentation and vocabulary inflation, which in turn influence the performance of language models trained on comparable amounts of data.

A second and equally important aspect is that tokenization originates in data compression. Byte Pair Encoding (BPE), now ubiquitous in NLP, was first introduced by Gage (1994) as a general-purpose compression method. The algorithm iteratively replaces the most frequent pair of adjacent bytes with a new symbol and stores the mapping in a lookup table; modern tokenizers adopt the same strategy but stop after a fixed number of merges to obtain a desired vocabulary size. More recently, Goldman et al. (2024) show that the compression capacity of a tokenizer correlates strongly with downstream model performance: tokenizers that reduce entropy more effectively tend to yield better models, especially in low-resource settings. Seen through this lens, tokenization is not an afterthought but an intrinsic modeling decision that balances representation granularity, computational efficiency and information retention. Our experiments therefore compare not only frequency-based BPE but also linguistically informed alternatives, asking how different choices of basic units affect both compression and learning.

2.2 Tokenization as Morphological Segmentation

At the same time, tokenization is intimately related to the linguistic structure of words. Natural languages are compositional at multiple levels, and morphology provides some of the clearest evidence of this: words are built from smaller, meaningful units — morphemes — such as roots, inflectional markers, and derivational affixes. An ideal tokenization system would capture these junctures, segmenting text in a way that reflects its internal linguistic organization.

For example, while a frequency-based tokenizer might store *dog* and *dogs* as separate units, a morphologically-aware tokenizer would recognize that the plural form is derived from the singular by appending a regular inflectional morpheme *-s*. Segmenting at this level reveals productive patterns that aid generalization, allowing the model to infer the meaning and form of novel or rare words from their components.

Morphological segmentation has long been studied as a core component of linguistic theory and cognitive modeling. The psychological reality of morphemes is attested in experiments like the WUG test (Berko, 1958), where infants reliably extend morphological rules to novel forms.

These experiments indicate that children exhibit clear sensitivity to the distributional properties of morphemes in their linguistic input, an ability typically classified under statistical learning (Sandoval et al., 2017; Mehler et al., 1988). As learners internalize these patterns, they begin to abstract and generalize morphological rules following distinct trajectories, a phenomenon evidenced by systematic overgeneralization errors like *goed* for *went*, or *falled* for *fell* (Lignos and Yang, 2016). Indeed, the acquisition of morphological rules appears to follow what is known as the *Tolerance–Sufficiency Principle* (Yang, 2016), which provides a formal account of when a linguistic rule can be considered productive given the sparsity and irregularity of the input data. More concretely, the Tolerance Principle (TP) states that if a rule R may potentially apply to a set of N items, then R is productive if

and only if the number of exceptions e satisfies (1):

$$e \leq \theta_N \text{ where } \theta_N = \frac{N}{\ln N} \quad (1)$$

When the number of exceptions exceeds this threshold ($e > \theta_N$), the learner is expected to treat those cases as lexicalized, and the rule R is considered unproductive. A complementary formulation is provided by the *Sufficiency Principle* (SP), which specifies the minimum amount of evidence required to support an observed generalization. Formally, given a generalization R over N items, where R is attested in M cases, R is extended to the remaining $N - M$ items if and only if:

$$N - M \leq \theta_N \text{ where } \theta_N = \frac{N}{\ln N} \quad (2)$$

The question, then, is whether artificial systems can similarly benefit from identifying morphemes — and whether doing so would support better compression, generalization, and interpretability. Subword tokenization (e.g., Byte-Pair Encoding) is in fact able to capture recurring internal structure within words, such as prefixes, suffixes, and roots (e.g., *un-* + *believ* + *able*), allowing the model to generalize across unseen words. This can be viewed as a form of compositional representation that mirrors the generative flexibility of human morphology. However, unlike morphemes, LLM tokens are not guaranteed to be semantically meaningful, and their segmentation is only driven by frequency optimization rather than grammaticality or communicative function. A truly cognitively plausible tokenizer is unlikely to achieve optimal efficiency as defined purely by compression or predictive performance. Instead, it might display the kinds of overgeneralization errors and irregularities that characterize child language acquisition. Such "imperfect" segmentation reflects the underlying learning process rather than a finished, fully optimized system.

2.3 Toward an Integrated Perspective

The three perspectives above—tokenization as modelling, compression and morphological segmentation—are complementary rather than competing. They suggest that linguistic plausibility, information-theoretic efficiency and engineering convenience can, in principle, be aligned. To put this hypothesis to the test, we explore a diverse family of tokenizers designed to identify basic linguistic units (syllables and morphemes) as tokens.

We evaluate these tokenizers along two complementary axes. First, we assess each tokenizer on its own by measuring how well it segments words into morphemes using a re-adapted version of the SIGMORPHON 2022 morpheme-segmentation benchmark (Batsuren et al., 2022), quantifying their morphological "soundness." Second, we pair each tokenizer with a fixed GPT-2 architecture and train on the BabyLM 2025 strict-small corpus. The resulting models are evaluated on the BabyLM challenge suite of tasks—ranging from linguistic preference tests (BLiMP, BLiMP-Supplement, EWoK) to downstream fine-tuning (GLUE)—as described by Warstadt et al. (2023b). By correlating segmentation quality and model performance, we aim to clarify whether linguistically motivated tokenizations lead to tangible benefits for small-scale language modelling.

The next sections build on this motivation. Section 3 surveys prior work on unsupervised morphology, paradigm discovery and the role of tokenization in language modelling, providing the theoretical context for our tokenizer designs. Section 4 details the datasets, tokenizer construction and evaluation used in our experiments, followed by an analysis of results across both morphological and BabyLM benchmarks.

3 Related Work

3.1 Unsupervised morphological segmentation

Morphological segmentation has long been viewed as both a descriptive and an information-theoretic problem. Goldsmith (2001) introduced the *Linguistica* system, framing the discovery of morphemes and paradigms as a minimum description length (MDL) optimization task. By balancing model complexity against data fit, the algorithm learns a lexicon and a set of affix patterns that jointly minimize description length, using these patterns to predict segmentation points in unseen words. Subsequent work by Xu et al. (2018) proposed a probabilistic model that identifies roots, suffixes, and transformation rules to generate candidate segmentations for each word, and then induces shared paradigms to filter out spurious affixes. Both studies demonstrate that paradigm extraction is critical for capturing the combinatorial nature of morphology; this insight motivates our morphology-oriented tokenizers, which aim to discover recurring patterns rather than simply splitting words into arbitrary subword units.

3.2 Tokenization and compression in language modelling

Subword methods such as byte-pair encoding (BPE) are now ubiquitous, yet recent research has emphasized the deeper role tokenization plays in language modelling itself. Goldman et al. (2024) systematically investigate how different tokenizers affect model performance through the lens of text compression, showing that tokenizers with lower empirical entropy (i.e., greater compression) tend to yield better downstream performance. They argue that tokenization should be viewed as an integral component of the modelling pipeline rather than as a mere preprocessing step. Fusco et al. (2024) first proposed MorPiece, a tokenization strategy based on a Trie structure for the representation of the entire lexicon, identifying splits though the application of the Tolerance–Sufficiency Principle (Yang, 2016) (see also Section 4.2). Oh and Schuler (2025) examine how token granularity influences the predictive power of language models' surprisal measures relative to human processing data. Bunzeck et al. (2025) compare grapheme- and phoneme-based small models, finding that they perform comparably to their subword analogues trained on the same limited token budget. Pagnoni et al. (2025) introduce an LLM architecture that eliminates tokenization altogether, instead representing text as variable-length byte patches defined dynamically by entropy. Finally, Raj S et al. (2025) employ a Viterbi-like algorithm that also operates on a Trie-based representation of the vocabulary to compute globally optimal segmentations, reporting improvements over the greedy BPE baseline on both intrinsic and extrinsic metrics.

3.3 Morphological segmentation as a shared task

The SIGMORPHON series of shared tasks provides a valuable benchmark for evaluating morphological models. In the 2022 edition of the morpheme segmentation task, Batsuren et al. (2022) challenged systems to decompose words and sentences into sequences of morphemes across a diverse set of languages. Subtask 1 comprised a 5 million-word corpus covering nine languages, while subtask 2 involved sentence-level segmentation in three languages. The best systems achieved an average F_1 score of 97.3 % across languages and outperformed standard tokenizers such as BPE and Morfessor by more than 30 percentage points. These results demonstrate that data-driven morphological segmenters can capture complex derivational and inflectional patterns and that morphological segmentation yields more accurate boundaries than generic subword methods. We adapt the SIGMORPHON 2022 data for evaluating our tokenizers, using its gold-standard segmentations to quantify how well each tokenizer preserves morphological structure.

3.4 Implications for BabyLM and our study

Previous BabyLM submissions highlight the importance of model architecture and input representation. The GPT-BERT model of Charpentier and Samuel (2024) shows that combining masked and causal objectives can improve BabyLM scores by enabling a single transformer to operate in both modes. The phoneme-based approach of Goriely et al. (2024) demonstrates that non-standard tokenizations (phonemic transcription, character-level segmentation) can yield competitive performance, albeit with trade-offs such as slight drops on text-based tasks. Our work continues this tradition by systematically comparing GPT-2 models trained with multiple linguistically oriented tokenizers—including BPE, character-level, morphology-based and hybrid variants—and relating their BabyLM performance to their ability to segment words morphologically. By bridging insights from unsupervised morphology, tokenization compression theory and SIGMORPHON evaluations, we aim to better understand how tokenization choices shape the learning and generalisation of small language models.

4 Experiments

In this section we describe the data, tokenizers, model configurations and evaluation protocols used in our study. Wherever possible, we follow the BabyLM challenge guidelines to ensure comparability with prior work.

4.1 Data and Preprocessing

BabyLM corpus. Our training data come from the BabyLM 2025 strict-small track, which offers a fixed corpus of roughly 10M words. The dataset comprises text from six distinct domains that reflect diverse linguistic contexts. Data are taken from conversational sources, with CHILDES contributing 29% child-directed dialogue data and OpenSubtitles providing 20% scripted dialogue, while written materials include Project Gutenberg's fiction

and nonfiction works (26%) and Simple English Wikipedia entries (15%). Additional dialogue data comes from the British National Corpus (8%) and Switchboard telephone conversations (1%). We applied a lightweight preprocessing, consisting of space normalization, lowercasing and separation of alphabetic characters from digits and punctuation, except for apostrophes.

SIGMORPHON segmentation benchmark. In order to perform an *intrinsic* evaluation of our tokenizers, independently of the language model, we turn to the word-level morpheme segmentation dataset released as part of the SIGMORPHON 2022 shared task on morpheme segmentation. The organisers provided gold segmentations for nine languages (Czech, English, Spanish, Hungarian, French, Italian, Russian, Latin and Mongolian) and reported that the best systems achieved an average F_1 score of 97.3 % across languages. Importantly, the task distinguishes between two kinds of segmentation. The original "deep" or *canonical* segmentation aligns each segment with an underlying lemma; morphemes are restored to their canonical shapes even if surface forms have undergone phonological or orthographic changes. For example, the English noun *collision* is canonically segmented as *collide+ion*, rather than its surface segmentation *collis+ion*; likewise, *profitably* would be segmented as *profit+able+-ly*. This canonical segmentation is "deeper" in that it recovers latent morphological structure beyond simple boundary detection, effectively lemmatising each morpheme.

In this work we focus on a more practical, "shallow" segmentation that is closer to tokenization. We convert the canonical segmentations provided in the SIGMORPHON test set into surface-level boundaries by simply inserting split markers at morpheme boundaries without altering the character sequence. That is, we segment *collision* as *collis+ion* and *profitably* as *profit+ably*, leaving the surface text unchanged. This conversion (i) avoids introducing extra graphemes or lemma forms that would not appear during training, (ii) aligns the task with tokenization, where the goal is to identify basic units in the input rather than to normalise them, and (iii) is theoretically motivated by the view that morphological composition operates over both roots and affixes, so a tokenizer should aim to segment wherever composition occurs—even if the base form shows no internal change. The difference between deep and shallow segmentation is less pronounced in English than in languages with richer fusional morphology, but the shallow version still provides a useful proxy for measuring how well a tokenizer captures morpheme boundaries. In our experiments we extract only the English portion of the SIGMORPHON test set and use the resulting shallow segmentation as a gold standard for evaluating each tokenizer's morphological soundness (Section 4.4).

4.2 Tokenizers

For our experiments we compare the standard byte-pair encoding (BPE) tokenizer (Gage, 1994; Sennrich et al., 2016) against three linguistically motivated tokenizers: *MorPiece* (MoP) (Fusco et al., 2024), *SylliTok*, and *ParadigmFinder* (ParFind). All tokenizers are trained on the BabyLM 10M training data with a maximum vocabulary size of 30 000 tokens, using identical preprocessing. Below we summarise the key design principles of each.

MorPiece (MoP). MorPiece segments words into morpheme-like units by postulating a split whenever the Tolerance–Sufficiency Principle (SP) (Yang, 2016) can be applied during the traversal of the lexicon. The current implementation diverges minimally from the original MoP model (Fusco et al., 2024) while preserving its Trie-based lexical structure. Consider the word $cats$: a root Trie ($c \to a \to t \to s$) and an inflectional Trie ($s \to t \to a \to c$) are created. "Traversing" the lexicon entails incrementing by one the counter of each node encountered in both the root and inflectional tries. If a path does not exist, it is assigned an initial value (i.e., frequency) of one. A split between t and s is postulated if, and only if, the SP is satisfied in both the root Trie and the inflectional Trie, that is:

$$\text{split } \textit{iff} \text{ in } \textit{root-trie}\text{: } \frac{\text{freq}(t)}{\ln(\text{freq}(t))} > \text{freq}(s) \qquad \text{and in } \textit{infl-trie}\text{: } \frac{\text{freq}(s)}{\ln(\text{freq}(s))} > \text{freq}(t)$$

If this is the case, the s pendant (in this instance, simply s) is added to the root Trie, rather than to the special root node "++" as in the original MoP. At the end of processing, all nodes with a frequency below the *min_freq* parameter are pruned. A MaxLength strategy is then applied to retrieve the tokens for each word during encoding.

In our experiments, the training procedure was constrained to prune the dictionary according to the *min_freq* parameter and to save a vocabulary version every 100K tokens of exposure. The resulting

vocabularies are useful for verifying each splitting hypothesis and the evidence required to postulate it (with the order of acquisition, *ooa* parameter, set to *True*).

During vocabulary construction, the algorithm traverses the Trie and identifies candidate segmentation points according to two hyper-parameters: a *cutoff* threshold and a *branching factor* (*bf*). The *cutoff* specifies the minimum frequency a mother node must reach before a split is postulated (set to 100 in all our experiments). The *branching factor* specifies the minimum number of distinct daughters a mother node must have in order for the SP to apply (set to 2 in all our experiments).

Unlike BPE, MorPiece relies on linguistic cues—high type frequency and morphological variability—to determine split points. It does not depend on precompiled morpheme lists (as in (Jabbar, 2023)), but instead induces potential morphemes directly from the data using the trie. The settings we adopted favor plausible segmentations and capture frequent inflectional and derivational affixes (e.g., *-ed*, *-ing*, *-s*, *-ness*, *un-*) while preventing oversegmentation of rare strings. However, the current vocabulary-building procedure does not delete a pendant when a split is postulated; pendants are removed only during the pruning step of the optimization phase. For this reason, we applied an aggressive optimization strategy, pruning nodes below the *min_freq* threshold every 100K tokens of exposure. This process yielded a vocabulary of approximately 23K tokens under the *strict-small* training regime and about 40K tokens under the *strict* training regime.

SylliTok. SylliTok is a rule-based tokenizer designed to align token boundaries with the syllabic structure of English. Linguistic and psycholinguistic research has shown that infants are highly sensitive to syllable-level patterns in continuous speech, often segmenting syllables before larger morphological units. Building on this insight, SylliTok uses deterministic syllabification rules to split words into syllables, yielding a token vocabulary of size 20K. For example, *banana* is tokenised as *ba-na-na* and *computer* as *com-pu-ter*. In languages with relatively transparent orthography, such as Spanish and Italian, the mapping from orthography to syllables is straightforward; in English it is more complex due to inconsistent spelling–sound correspondence. Nonetheless, a syllable-based tokenizer provides a cognitively plausible baseline and reduces token length in a way that may benefit low-resource models.

ParadigmFinder (ParFind). ParFind is another unsupervised tokenizer that extracts paradigms from the vocabulary and uses them to segment words, following previous work by Goldsmith (2001) and Xu et al. (2018). In our framework, a *paradigm* consists of a set of roots and a corresponding set of suffixes that co-occur with systematic regularity. For example, the words *walk*, *walks*, *walked* and *walking* are evidence for a paradigm with root *walk* and suffixes {`-∅`, `-s`, `-ed`, `-ing`}. ParFind induces such paradigms from the data in a multi-step process. The search is initialized by enumerating all possible binary splits of words into candidate roots and suffixes, and then grouping together roots that share identical suffix sets (see Fig. 1). The algorithm then normalizes suffixes by factoring out common prefixes and appending them to the roots, ensuring that segmentation points correspond to genuine morphological variation in at least one case in each paradigm. This step prevents the formation of spurious paradigms such as {`-t`, `-ts`, `-ted`, `-ting`}, which arise when several verb roots share the final letter (e.g., *-t*) (see also Goldsmith, 2001 on this issue). At this point, paradigms are expanded according to a Tolerance–Sufficiency Principle, enabling generalization to unseen forms. Formally, given two paradigms P_i and P_j with root sets R_i and R_j and corresponding suffix sets S_i and S_j, where $|S_i| < |S_j|$, P_i is merged into P_j if and only if a majority of roots in R_j occur in R_i, that is, if

$$|R_j| - |R_i \cap R_j| \geq \theta_{|R_j|} \quad \text{where} \quad \theta_{|R_j|} = \frac{|R_j|}{\ln |R_j|} \tag{3}$$

This condition ensures that paradigms are merged only when the overlap between root sets provides sufficient evidence for systematic extension rather than accidental co-occurrence. When words can still be analyzed with different segmentations according to multiple paradigms, suffixes are checked against existing paradigms to determine whether nested suffixation is possible. For instance, the words *singer* and *singers* can be segmented as *singer-∅* and *singer-s* under paradigm P_1, and as *sing-er* and *sing-ers* under paradigm P_2. In this case, since P_1 is already a productive paradigm, *-er* and *-ers* are in turn analyzed as *-er-∅* and *-er-s*. Finally, the algorithm prunes redundant or subsumed

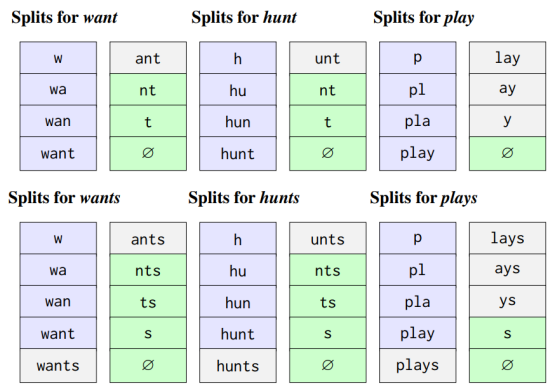


Figure 1: All possible binary splits for *want*, *hunt*, *play*, *wants*, *hunts* and *plays*. Boxes in purple indicate recurring roots, boxes in green stand for recurring suffixes. The paradigm that best accounts for the six lexical items is the one formed by productive roots *want*, *hunt* and *play*, and the productive suffixes -∅ and-*s*.

paradigms and ranks the remaining ones using a support score (adapted from Goldsmith, 2001): for any paradigm P with root set R and suffix set S, the support score is defined as

$$\text{Score}_P = \log_2(|R|) \times \log_2(|S|). \quad (4)$$

Words that do not belong to any paradigm are assigned to a "residual" paradigm, preventing spurious segmentations. During tokenization, ParFind first attempts to match a word against known paradigms and segment it accordingly; if no exact match is found, a fallback strategy matches the longest known suffix to recover partial structure.

All roots and suffixes from the paradigms, including those of the residual paradigm, are assigned unique token IDs. The vocabulary size obtained through this procedure was explicitly set to 30K.

4.3 Model Architecture and Training

We use GPT-2 as our base model to align with the BabyLM baselines and previous submissions. Unless stated otherwise, we train separate models for each tokenizer in both the strict and strict-small tracks.

Architecture. Our GPT-2 implementation follows the "base" configuration with 12 transformer layers ($n_{\text{layer}} = 12$), hidden size $n_{\text{embd}} = 768$, and 12 self-attention heads ($n_{\text{head}} = 12$). Each model has a context window of $n_{\text{positions}} = 1024$ tokens. This architecture yields approximately 110M trainable parameters. For fair comparison across tokenizers, we keep the non-embedding parameters fixed and adjust only the input embedding layer to accommodate the vocabulary size of each tokenizer.

Training procedure. Models are trained using the official BabyLM recipe. We adopt the following hyper-parameters:

- **Sequence length**: 512 tokens per example.
- **Batch size**: 16 examples.
- **Optimiser and learning rate schedule**: AdamW with a base learning rate of 5×10^{-5}, linear warm-up over the first 2,000 steps and weight decay of 0.01.
- **Training steps**: 200,000 steps (roughly 10 epochs over the strict-small data).
- **Gradient clipping**: max norm of 1.0.

4.4 Results

We evaluate both the tokenizers and the trained language models.

Morphological segmentation. For each tokenizer, we segment the SIGMORPHON benchmark words and compute precision, recall, F_1 and Levenshtein distance against the gold morpheme boundaries, following the SIGMORPHON 2022 evaluation procedure. These scores allow us to quantify how well each tokenizer captures linguistically meaningful units.

Results are shown in Table 2. The best performance on this benchmark is reached by ParadigmFinder, with an F1 score of 33.99, followed by MorPiece (F1 = 26.80), BPE (F1 = 23.50) and finally SylliTok (F1 = 14.98). This ordering is consistent with our expectations: both ParadigmFinder and MorPiece explicitly target morphemes as the fundamental units of segmentation, albeit in fully unsupervised ways, and therefore align more closely with the gold morphological boundaries. In contrast, BPE optimises for compression rather than linguistic structure, and SylliTok splits on syllables, a unit that often does not coincide with morpheme boundaries in English.

Tokenizer	Avg. Lev. Dist.	Prec	Rec	F1
BPE	2.08	21.03	26.62	23.50
MoP	1.96	24.54	29.52	26.80
SillyTok	2.77	12.45	18.81	14.98
ParFind	**1.24**	**38.99**	**30.12**	**33.99**

Table 1: Tokenizers' evaluation on SIGMORPHON using BabyLM 10M as training corpus.

BabyLM evaluation. We use the official BabyLM 2025 evaluation pipeline to assess our custom tokenizers when paired with a standard GPT-2 model architecture. These tasks collectively probe a wide range of linguistic and cognitive abilities—from syntactic acceptability and morphological generalisation to world knowledge, entity state tracking and alignment with human reading behaviour—providing a comprehensive evaluation of our tokenizers and models. We refer to the Appendix for a detailed description of the various tasks.

In Table 2, we report the macro-averaged score for each section of the benchmark. We compare the performance of our models with that of the challenge's baseline GPT-2 model with the BPE tokenizer. The results show no single tokenizer dominating across all tasks, but rather a complementary pattern that reflects the different linguistic biases each segmentation strategy encodes.

SylliTok performs surprisingly well on comprehension-oriented tasks: it achieves the best scores on BLiMP Supplement (58.8) and GLUE (58.1), while matching BPE performance on EWoK (≈49.9). However, it shows only a weak positive correlation with human judgements on WUG_ADJ (33.1) and a negative correlation on WUG_PAST (-29.4), highlighting the limits of a syllable-based representation when it comes to morphosyntactic generalisation.

MorPiece, which segments words into morphemes, offers a different trade-off: it improves semantic and discourse tasks—scoring higher on COMPS (55.8), EWoK (50.6) and by far the best on Entity Tracking (64.4) and WUG_PAST (12.1)—but it trails BPE on BLiMP and BLiMP Supplement and yields weaker results on WUG_ADJ (37.6) and AoA (-25.6), suggesting that morphological segmentation alone does not uniformly translate to improved performance.

ParadigmFinder achieves the best scores on the semantic task COMPS (56.6) and on BLiMP Supplement, matching SylliTok performance (58.8). It also yields the best AoA score (16.3), indicating a degree of alignment with developmental learning patterns of words. The surprisingly low performance on WUG_ADJ (-43.1) may be attributed to the difficulty of recognising multiple derivational suffixes.

BPE, while linguistically shallow, remains a strong baseline. It leads on BLiMP (66.4) and WUG_ADJ (66.1), showing that purely frequency-driven segmentation can sometimes outperform more linguistically motivated methods. Nonetheless, its generally moderate scores across the other tasks confirm that frequency alone is insufficient to consistently capture the kinds of regularities targeted by BabyLM with a small token budget (10M).

5 Discussion

The results indicate that introducing linguistically informed tokenizers does not lead to clear improvements on the more traditional grammar-oriented sections of the BabyLM benchmark. On BLiMP, for instance, all models perform at a similar level, with BPE in fact yielding the highest score. Likewise, on BLiMP Supplement, the differences are small, with ParadigmFinder and SylliTok only slightly surpassing BPE. This suggests that morphologically and syllable-aware tokenizations do not provide systematic advantages on syntactic acceptability judgments, at least under the strict 10M training budget.

More interestingly, gains appear in tasks that require richer semantic generalisation and discourse tracking. Both SylliTok and ParadigmFinder equate BPE on EWoK, while MorPiece slightly outperforms it. All of our models also surpass it on COMPS and Entity Tracking, pointing to improvements in comprehension-oriented evaluation. In particular, MorPiece achieves the highest score on Entity Tracking, highlighting the potential of morpheme-based segmentation for tasks that demand sensitivity to discourse-level dependencies. ParadigmFinder, on the other hand, shows competitive results on semantic composition (COMPS) and also exhibits a modest advantage in word Age of Acquisition (AoA), suggesting that a paradigm-based segmentation may capture aspects of lexical development more effectively than frequency-based subword units.

These results align with the findings from the intrinsic evaluation on the SIGMORPHON segmentation benchmark. There, ParadigmFinder and MorPiece achieved the best correspondence to morpheme boundaries, while BPE and SylliTok lagged behind. The parallel between segmentation accuracy and downstream comprehension/discourse gains suggests that morphological faithfulness in tokenization may indeed translate into advantages for meaning-sensitive tasks, even if not for purely grammatical ones.

Model	BLiMP	BLiMP Suppl.	COMPS	EWoK	Eye Track.	SPR	Entity Track.	WUG_ADJ	WUG_PAST	GLUE	AoA
GPT-2 + BPE	**66.4**	57.1	51.7	49.9	**8.7**	**4.3**	13.9	**66.1**	-5.0	55.9	11.7
GPT-2 + MoP	63.5	52.6	55.8	**50.6**	1.2	0.7	**64.4**	37.6	**12.1**	57.7	-25.6
GPT-2 + SylliTok	63.1	**58.8**	55.3	49.9	0.9	0.1	33.9	33.1	-29.4	**58.1**	-31.7
GPT-2 + ParFind	65.2	**58.8**	**56.6**	49.4	0.1	0.3	21.0	-43.1	-2.6	57.8	**16.3**

Table 2: Results of the BabyLM tasks evaluation of the baseline GPT-2 model trained using different tokenization strategies.

6 Conclusion

In this work, we evaluated several tokenizers designed to approximate linguistic units and tested them both in isolation (via the SIGMORPHON 2022 morpheme segmentation benchmark) and when paired with GPT-2 on the BabyLM 2025 evaluation suite. The findings indicate that while linguistically motivated tokenizers do not consistently outperform BPE on grammar-focused benchmarks, they offer complementary benefits on tasks targeting comprehension, discourse tracking, and developmental plausibility.

Taken together, the results reinforce our threefold perspective on tokenization: **(i) modeling**, since the segmentation choice directly affects the inductive biases available to the language model; **(ii) compression**, since different strategies vary in how efficiently they reduce entropy and distribute representational resources; and **(iii) morphology**, since tokenization determines the extent to which models can access and exploit the systematic structure of words. The SIGMORPHON results demonstrate that more morphology-aware tokenizers are indeed closer to humanlike segmentation, and the BabyLM evaluation reveals that this morphological consistency carries over into improvements in meaning-sensitive tasks.

Future work should expand evaluation to multiple languages, integrate hybrid tokenization strategies, and further investigate the alignment between human morphological acquisition and artificial segmentation methods. Ultimately, our findings suggest that tokenization should be treated not as a fixed preprocessing step, but as a substantive modeling decision with theoretical and practical consequences.

7 Limitations

Our experiments were conducted on the *strict-small* BabyLM corpus (10K tokens) rather than the full *strict* version (100K tokens). A direct comparison with models and tokenizers trained on the larger corpus would be essential to assess how data scale influences both tokenization quality and downstream performance.

Furthermore, we restricted our evaluation to a single baseline architecture (GPT-2). While this choice allowed for controlled comparisons across tokenization strategies, future work should test the generality of our findings across models of different sizes and architectures.

Finally, the relation between tokenization and compression remains to be explored in greater depth. In particular, future work should incorporate an explicit Minimum Description Length (MDL) metric (Goldsmith, 2001) to quantify how efficiently each tokenizer represents linguistic structure.

8 Ackowledgments

We acknowledge financial support under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.1, Call for tender No. 104 published on 2.2.2022 by the Italian Ministry of University and Research (MUR), funded by the European Union – NextGenerationEU – Project Title T-GRA2L: Testing GRAdeness and GRAmmaticality in Linguistics (202223PL4N) – CUP I53D23003900006 - Grant Assignment Decree No. 104 adopted on the 2nd February 2022 by the Italian Ministry of Ministry of University and Research (MUR). PI: CC. This work contains simulations carried out on the High Performance Computing DataCenter at IUSS, co-funded by Regione Lombardia through the funding programme established by Regional Decree No. 3776 of November 3, 2020

References

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Pho-*

netics, Phonology, and Morphology, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Jean Berko. 1958. The child's learning of english morphology. *Word*, 14(2-3):150–177.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2025. Small language models also work with small vocabularies: Probing the linguistic abilities of grapheme- and phoneme-based baby llamas. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6039–6048, Abu Dhabi, UAE. Association for Computational Linguistics.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. Gpt or BERT: why not both? *arXiv preprint*, arXiv:2410.24159.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Achille Fusco, Matilde Barbini, Maria Letizia Piccini Bianchessi, Veronica Bressan, Sofia Neri, Sarah Rossi, Tommaso Sgrizzi, and Cristiano Chesi. 2024. Recurrent networks are (linguistically) better? An (ongoing) experiment on small-lm training on child-directed speech in italian. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 382–389.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Omer Goldman, Avi Caciularu, Matan Eyal, Kris Cao, Idan Szpektor, and Reut Tsarfaty. 2024. Unpacking tokenization: Evaluating text compression and its correlation with model performance. In *Findings of the Association for Computational Linguistics: ACL 2024*. ArXiv:2403.06265.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Lisa Beinborn, and Paula Buttery. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. *arXiv preprint*, arXiv:2410.22906.

Valentin Hofmann, Leonie Weissweiler, David Mortensen, Hinrich Schütze, and Janet Pierrehumbert. 2024. Derivational morphology reveals analogical generalization in large language models. *arXiv preprint*. ArXiv:2411.07990.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R.T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *Transactions of the ACL*. ArXiv:2405.09605.

Haris Jabbar. 2023. Morphpiece: A linguistic tokenizer for large language models. *arXiv preprint arXiv:2307.07262*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, pages 3835–3855, Toronto, Canada.

Constantine Lignos and Charles Yang. 2016. Morphology and language acquisition. *The Cambridge handbook of morphology*, 743764.

Jacques Mehler, Peter Jusczyk, Ghislaine Lambertz, Nilofar Halsted, Josiane Bertoncini, and Claudine Amiel-Tison. 1988. A precursor of language acquisition in young infants. *Cognition*, 29(2):143–178.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. Comps: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)*, pages 2928–2949, Dubrovnik, Croatia.

Byung-Doh Oh and William Schuler. 2025. The impact of token granularity on the predictive power of language model surprisal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4150–4162, Vienna, Austria. Association for Computational Linguistics.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. Byte latent transformer: Patches scale better than tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.

Bharath Raj S, Garvit Suri, Vikrant Dewangan, and Raghav Sonavane. 2025. When every token counts: Optimal segmentation for low-resource language models. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 294–308, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Michelle Sandoval, Dianne Patterson, Huanping Dai, Christopher J Vance, and Elena Plante. 2017. Neural correlates of morphology acquisition through a statistical learning paradigm. *Frontiers in Psychology*, 8:1234.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023b. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, David Mortensen, and Janet Pierrehumbert. 2023. Evaluating morphological generalization with wug tests. *arXiv preprint*. ArXiv:230x.xxxxx.

Hongzhi Xu, Mitchell Marcus, Charles Yang, and Lyle Ungar. 2018. Unsupervised morphology learning with statistical paradigms. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 44–54, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Charles Yang. 2016. *The price of linguistic productivity: How children learn to break the rules of language*. MIT press.

A Appendix

The BabyLM 2025 benchmark is structured as follows:

Zero-shot linguistic preference tasks.

- **BLiMP** — The Benchmark of Linguistic Minimal Pairs (Warstadt et al., 2020) is a challenge set designed to probe what language models know about core grammatical phenomena in English. It comprises 67 automatically generated sub-datasets, each containing 1,000 minimal sentence pairs that isolate a particular syntactic, morphological or semantic contrast. Models are scored by whether they assign higher probability to the grammatical sentence in each pair.

- **BLiMP Supplement** — An extension of BLiMP tailored for BabyLM (Warstadt et al., 2023b). The supplement introduces additional contrasts (e.g. lexical and morphological judgments) not covered in the original BLiMP. As with BLiMP, models must prefer the acceptable sentence in each minimal pair.

- **EWoK** — The Elements of World Knowledge framework (Ivanova et al., 2025) evaluates basic world-modeling abilities by asking models to judge which of two context/target pairs is more plausible. The EWoK-CORE-1.0 dataset contains 4,374 items spanning 11 knowledge domains (from social interactions to spatial relations). Minimal pairs are constructed so that only one sentence aligns with commonsense world knowledge.

- **Entity Tracking** — Based on the task of Kim and Schuster (2023), this evaluation tests a model's ability to keep track of entities and their states as a text unfolds. A model is given an initial description of an entity and a series of state-changing operations and must assign higher probability to the correct continuation that reflects the entity's final state. In the BabyLM pipeline this task is evaluated in a zero-shot setting by computing sentence logit scores.

- **Derivational Morphology (WUG_ADJ)** — Following Hofmann et al. (2024), this task tests morphological generalisation via an adjective-nominalisation "wug" experiment.

Models see nonce adjectives (e.g. *daxen*) and must decide whether the corresponding noun uses the suffix *-ity* or *-ness*. Performance is measured by the correlation between model probabilities and human judgements.

- **WUG_PAST** — From Weissweiler et al. (2023), this hidden task evaluates how models generalise past-tense formation to nonce verbs. Models are presented with a novel verb and several possible past-tense forms; their probability distribution is correlated with human responses.

- **COMPS** — The Conceptual Minimal Pair Sentences dataset (Misra et al., 2023) tests whether language models know that properties of superordinate concepts are inherited by subordinate concepts. Sentences feature nonce words standing in hierarchical relations (e.g. a *lorp* is a type of *bim*); models must assign higher probability to the sentence that correctly inherits the property.

- **Cloze probability and reading time (Self-paced Reading and Eye-tracking)** — Adapted from de Varda et al. (2024), this benchmark links language model predictions to human reading times. The evaluation computes the increase in explained variance (R^2) in human eye-tracking measures with no spill-over effect and in self-paced reading with a one-word spillover. It assesses the alignment between model surprisal and human processing difficulty.

- **Age of Acquisition (AoA)** — Based on Chang and Bergen's (2022) methodology, this benchmark tracks word surprisal across training checkpoints to estimate when a model "acquires" each word. The resulting learning curves are fitted with sigmoid functions and correlated with human Age-of-Acquisition norms from the MacArthur–Bates Communicative Development Inventory.

Fine-tuning tasks.

- **(Super)GLUE** — The General Language Understanding Evaluation benchmark (Wang et al., 2018, 2019) comprises a suite of natural-language understanding tasks (e.g. sentiment analysis, paraphrase detection, natural-language inference). In BabyLM it is used to assess models' ability to generalise via supervised fine-tuning on tasks such as MNLI, SST-2, QQP and QNLI.

Batch-wise Convergent Pre-training: Step-by-Step Learning Inspired by Child Language Development

Ko Yoshida[1], Daiki Shiono[1], Kai Sato[1], Toko Miura [1], Momoka Furuhashi[1], Jun Suzuki[1,2,3],

[1]Tohoku University, [2]RIKEN, [3]NII LLMC,

Correspondence: yoshida.kou.p3@dc.tohoku.ac.jp

Abstract

Human children acquire language from a substantially smaller amount of linguistic input than that typically required for training large language models (LLMs). This gap motivates the search for more efficient pre-training methods. Inspired by child development, curriculum learning, which progresses from simple to complex data, has been widely adopted. In this study, we propose a pre-training framework that mirrors child language acquisition, advancing step by step from words to sentences while retaining prior knowledge. We investigate whether this improves retention and efficiency under limited resources. Our approach is implemented through four components: (i) a curriculum-aligned dataset, (ii) a batch-wise convergence loop, (iii) a distance-controlled loss to mitigate forgetting, and (iv) a constraint-controlled optimizer for stability. Experiments on the BabyLM benchmark show that the proposed method performs slightly below the official baselines in overall accuracy, with larger gaps on grammar-oriented evaluations such as BLiMP. Nonetheless, it yields small but consistent gains on morphology- and discourse-related tasks (e.g., WUG-ADJ, Entity Tracking), suggesting that the approach affects different linguistic aspects unevenly under limited data conditions.

1 Introduction

Recent language models (LMs) have achieved remarkable performance. They are typically trained on massive datasets, often containing trillions of tokens, which makes it difficult to attain comparable performance when only limited training data is available (Zhang et al., 2021; Kaplan et al., 2020). In contrast, human children are able to acquire language with far fewer words over their developmental period (Gilkerson et al., 2017), creating a substantial gap between human and machine learning efficiency (Dupoux, 2018). Bridging this gap requires developing training methods that can achieve strong language modeling performance even under limited data conditions. Such approaches are important not only for advancing natural language processing but also for providing insights into human language learning.

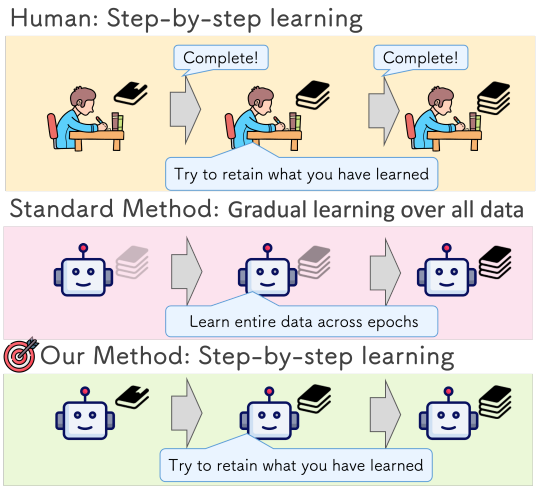


Figure 1: Inspired by human learning, our method trains language models on batches of gradually increasing difficulty while trying to retain knowledge from previous batches.

The BabyLM Challenge (Warstadt et al., 2023; Hu et al., 2024) was launched with this motivation in mind. Inspired by human language acquisition, it explores efficient pre-training methods under strict constraints on the amount of training data and computational resources.

Previous studies have proposed **Curriculum Learning** (Bengio et al., 2009) strategies grounded in studies of human language development. These approaches control the order in which training data is presented, starting with simpler linguistic constructs and then progressing to more complex ones. Difficulty has been defined by factors such as sentence length, punctuation count, vocabulary, syntactic complexity, and readability. Models trained under such curricula have shown improvements on

Proceedings of the First BabyLM Workshop, pages 520–536

some tasks compared to those trained on randomly ordered data (Hu et al., 2024).

In this study, we extend the concept of curriculum learning by combining it with a training algorithm that is designed to imitate the human step-by-step learning process, as illustrated in Figure 1. Specifically, we take inspiration from the way humans tend to master one concept before progressing to the next, while retaining previously acquired knowledge. To emulate this, our method ensures that the model continues training on a given batch until its loss falls below the pre-defined acceptable level before proceeding to the next one, thereby aiming to solidify retention and prevent forgetting of previously learned material.

Experimental results on the BabyLM benchmark show that our method does not yet match the performance of the official baselines, particularly on grammar-oriented tasks. However, it exhibits small but consistent advantages in morphology- and discourse-level evaluations, indicating that the proposed approach affects different linguistic aspects in distinct ways under limited data conditions.

To summarize, our contributions are as follows:

- We propose a novel training algorithm that aims to stabilize knowledge retention in LLMs by continuing training on each batch until its loss converges.
- We design a training curriculum that starts with short utterances of two to three words, extracted from child-directed speech, and progressively increases linguistic complexity, thereby mimicking stages of child language acquisition.

2 Related Work

2.1 Stages of Child Language Development

Language acquisition in children follows a gradual, stage-wise trajectory. Conti-Ramsden and Durkin (2012) provides a comprehensive overview, noting that vocabulary expands rapidly around ages one to two. By age two, children begin producing two-word combinations that gradually develop into longer, grammatically structured utterances.

In contrast, the standard training approach for large language models typically involves presenting the entire dataset at once, mixing sentences of varying complexity from the outset. This standard approach is computationally inefficient and does not reflect the incremental nature of knowledge acquisition observed in human learners.

Motivated by this discrepancy, our study proposes a curriculum that more closely mimics child language development. Instead of presenting complete sentences from the beginning, we introduce short utterances, typically two to three words as in child-directed speech incrementally, gradually increasing the linguistic complexity of the training data. This approach aims to emulate the cumulative learning process observed in early human language acquisition, where new knowledge builds upon previously acquired elements without discarding them.

2.2 Curriculum Learning Strategies for Language Models

Most curriculum learning strategies proposed in previous studies on LMs, including those in past BabyLM challenges (Warstadt et al., 2023; Hu et al., 2024) have focused on reordering existing text data based on measures of difficulty. For example, Capone et al. (2024) leveraged the observation that the first words learned by children are often highly concrete and perceptually grounded. They propose a curriculum based on lexical concreteness, categorizing the data into four stages from most to least concrete, and training the model on them sequentially.

Other approaches have used linguistic features such as sentence length, punctuation counts, or readability scores to estimate difficulty and sort the training dataset accordingly (Ghanizadeh and Dousti, 2024; Behr, 2024).

An exception is the study by Salhan et al. (2024), who explore a curriculum for masked language modeling by selectively masking different parts of speech at different stages of training. While this goes beyond simple data reordering, it still operates within the scope of conventional text and masking strategies.

Our curriculum starts with short utterances of around two to three words, as typically observed in child-directed speech, and then gradually increases linguistic complexity by expanding the utterances toward longer and more complex sentences. This approach mimics the developmental stages observed in child language acquisition, where vocabulary builds incrementally and previously acquired words are retained and reused as sentences grow in complexity.

2.3 Forgetting and Lifelong Learning

Humans and animals are capable of continuously acquiring knowledge and skills throughout their lives. However, when neural networks are trained sequentially on data drawn from different distributions, the incremental acquisition of new information generally leads to catastrophic forgetting or interference with previously acquired knowledge (French, 1999).

Research on Lifelong Learning has been extensively discussed in the context of neural networks, and a comprehensive review is provided by Parisi et al. (2019). Various approaches have been proposed to mitigate forgetting, including replay-based methods that reuse past examples, regularization-based methods that constrain parameter updates, and architectural methods that expand the network with additional neurons or layers, among others.[1]

Curriculum learning improves efficiency by ordering training data, and when combined with Lifelong Learning techniques for mitigating forgetting, it can form a framework that both accelerates learning and preserves knowledge. In this study, we integrate curriculum learning with a batch-wise regularization mechanism that constrains parameter deviation while ensuring loss convergence. To our knowledge, this integration represents a novel direction in the context of pre-training small-scale language models such as BabyLM.

3 Method

Inspired by the stepwise nature of human language acquisition, we implement the learning dynamic of *advancing to the next batch only after sufficiently mastering the current one* as a constrained optimization problem. At each step t, we minimize a distance term that limits deviation from the previous parameters $\boldsymbol{W}^{(t-1)}$ while enforcing, as a *constraint*, that the cross-entropy loss on the current batch $\mathcal{X}^{(t)}$ meets a prescribed criterion. We solve this sequentially for $t = 1, 2, \ldots$, which jointly targets (i) *mastery before progression* within each batch, (ii) retention of previously acquired knowledge, and (iii) stability under small-data conditions.

We first summarize the pre-training data and curriculum (Section 3.1), and then outline the learning framework, including its *formulation* and *optimization* (Section 3.2). We describe a consolidation step based on parameter averaging that further reduces forgetting (Section 3.3). Finally, Implementation-specific choices—such as stopping rule constants, step-size clipping, hyperparameter listings, and the concrete checkpoint-averaging protocol—are summarized in Section 3.4.

[1] See Appendix A for more details on each approach.

Category	Words	Share (%)
Short-utterance data	3,824,058	3.824
Medium-utterance data	3,399,252	3.399
Synthetic short-sentence data	12,668,607	12.669
Narrative & dialogue data	80,107,855	80.108
Total	999,999,772	100.00

Table 1: Word counts of the pre-training dataset. Each category's proportion of the total is shown in the "Share" column.

3.1 Pre-training Data and Curriculum

3.1.1 Design goal.

To emulate the developmental trajectory from short utterances to full sentences and narratives, we tailor both the *data* and its *presentation schedule*. Our pre-training set integrates four sources of increasing complexity: two to three words utterances extracted from child-directed speech, slightly longer short utterances from the same source, synthetic short sentences, and finally longer child-directed narratives and dialogues. The curriculum exposes these sources in a gradually increasing order of linguistic complexity.

3.1.2 Datset components.

We compose the dataset from four sources, designed to support a smooth progression from lexical to sentential learning shown in Figure 2.

Short-utterance data. From the CHILDES dataset (Macwhinney, 2000) in the official BabyLM corpus (Choshen et al., 2024), we extract child-directed utterances of two to three words after removing metadata and non-linguistic markers. These short utterances form the starting point of our training material, corresponding to the earliest stage of child-directed input.

Medium-utterance data. We extract four to ten word utterances from CHILDES using the same preprocessing for short-utterance data. These examples extend minimal expressions and serve as the next stage of training material.

Synthetic short-sentence data. From English AoA ratings (Kuperman et al., 2012), we select

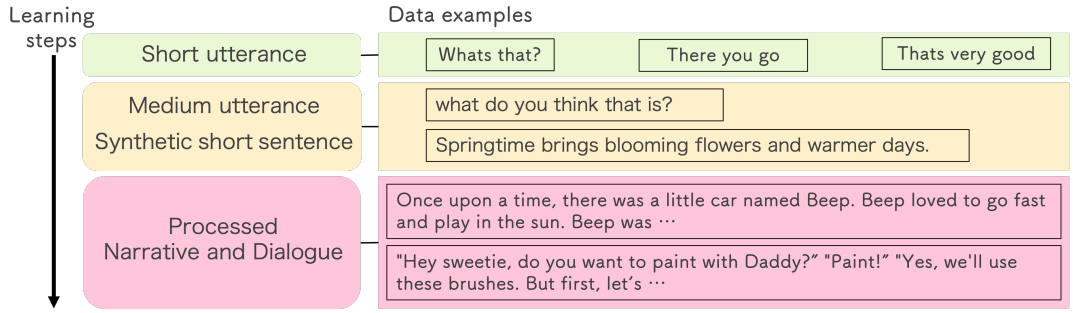


Figure 2: Overview of the staged curriculum used to construct our dataset. It progresses from short to medium CHILDES utterances, then to LLM-synthesized short sentences, and finally to processed narratives and dialogues, supporting a gradual transition from minimal expressions to full discourse.

words typically acquired by age 13 as lexical stimuli for the initial stage. For each word, Qwen3-14B (Team, 2025) generates one to fifteen word sentences showing simple literal uses while excluding named entities and complex syntax. Sentences exceeding the length cap are removed, and markup and punctuation normalized, yielding concise exemplars bridging lexical and phrasal structures.

Processed narrative and dialogue data. We preprocess three child-directed sources—KidLM (Nayeem and Rafiei, 2024) (essays), TinyStories (Eldan and Li, 2023) (narratives), and TinyDialogues (Feng et al., 2024) (dialogues)—to control length and remove extraneous markup. We shorten overly long sentences (splitting or compressing while preserving meaning), remove tags and metadata, and compress redundantly verbose paragraphs. This yields streamlined narrative/dialogue material that remains semantically coherent and aligns with our staged curriculum.

3.1.3 Two-axis curriculum.

We orchestrate the presentation order along two complementary axes.

In-batch curriculum (mixture scheduling). Within each batch, data are sampled from four sources with mixture weights that evolve during training. The share of short utterances starts high and steadily declines, medium utterances decrease more slowly, synthetic short sentences form a mid-stage peak, and narratives/dialogues increase toward the end. Let $\tau \in [0, 1]$ denote normalized training progress; smooth piecewise-linear schedules $\omega_{\text{short2-3}}(\tau)$, $\omega_{\text{short4-10}}(\tau)$, $\omega_{\text{synthetic}}(\tau)$, and $\omega_{\text{long}}(\tau)$ satisfy $\sum \omega = 1$, with early-stage data decreasing, late-stage data increasing, and intermediate data forming a unimodal peak.

Through-batch curriculum (difficulty ordering). Across batches, items are sorted by estimated difficulty: CHILDES-derived short and medium utterances appear first, synthetic sentences follow by length and lexical rarity, and narratives/dialogues by a composite difficulty score(see Appendix C).

3.2 Learning Framework

We conceptualize pre-training as a sequence of batch-wise learning problems. For each batch, the model is updated repeatedly until convergence, then proceeds to the next. Each update minimizes loss while constraining large parameter deviations from the previous state. This design aims to consolidate knowledge within each batch and retain prior learning, enabling gradual, stage-wise acquisition.

3.2.1 Batch-wise Convergent Pre-training Procedure

We now present the principle and motivation of the proposed *batch-wise convergent pre-training* loop.

Let $t \in \{1, 2, \ldots, T\}$ be the time step of the pre-training process. Let $\mathcal{X}^{(t)}$ denote the t-th subset of the training data, which we usually refer to as a *batch*. Let $\phi(\boldsymbol{W}, \mathcal{X}^{(t)})$ denote the cross-entropy over the t-th batch $\mathcal{X}^{(t)}$ between the given target probability distribution $\boldsymbol{p}(x)$ and the model's distribution $\boldsymbol{q}(\boldsymbol{W}, x)$ parameterized by $\boldsymbol{W}$, where $x \in \mathcal{X}^{(t)}$:

$$\phi(\boldsymbol{W}, \mathcal{X}^{(t)}) = -\frac{1}{|\mathcal{X}^{(t)}|} \sum_{x \in \mathcal{X}^{(t)}} \boldsymbol{p}(x) \log\big(\boldsymbol{q}(\boldsymbol{W}, x)\big). \tag{1}$$

Note that $\phi(\boldsymbol{W}, \mathcal{X}^{(t)}) \geq 0$ holds.

We then iterate *inner updates* until $\phi(\boldsymbol{W}, \mathcal{X}^{(t)})$ satisfies a threshold ε. Conceptually, this corresponds to minimizing the following sequence of

constrained optimization problems:

$$\boldsymbol{W}^{(t)} = \arg\min_{\boldsymbol{W}} \left\{ \frac{C}{p} \|\boldsymbol{W} - \boldsymbol{W}^{(t-1)}\|_p^p \right\} \quad \text{subject to:} \quad \phi(\boldsymbol{W}, \mathcal{X}^{(t)}) \leq \varepsilon, \tag{2}$$

where p, C and ε are hyperparameters, and $\boldsymbol{W}^{(0)}$ is the initial parameter matrix. Moreover, p represents the L_p-norm and $C \geq 0$ controls the strength of the distance term. Note that we only consider the $p \in \{1, 2\}$ cases. This formulation explicitly encodes the principle of *learning the current batch with the smallest necessary parameter change*.

Interpreting the threshold. ε corresponds to the negative log of a target average token accuracy (e.g., $-\log 0.5 = 0.6931$, $-\log 0.9 = 0.1054$), which provides a principled grounding of the constraint in probabilistic terms. This interpretation ensures that ε is not chosen arbitrarily but reflects a meaningful accuracy level; implementation-specific instantiation is deferred to Section 3.4.

Intuition and benefits. Single-pass training, with one update per batch, may progress while leaving content unlearned, causing oscillation between forgetting and relearning. In contrast, our *progress-upon-attainment* criterion ($\phi \leq \varepsilon$) promotes stage-wise consolidation. Together with distance control described below, updates are confined to the *minimum necessary change* to reach the target.

Role of the distance term. With $p = 2$, the term induces smooth *contraction* toward $\boldsymbol{W}^{t-1}$; with $p = 1$, it creates per-parameter *dead bands* (i.e., effective freezing zones of width C) around $\boldsymbol{W}^{t-1}$. The former ensures stability, while the latter selectively immobilizes less important parameters.

3.2.2 Constrained Objective and Lagrangian Formulation

Building on the criterion introduced in the previous section, at each step t, our update solves a sequential constrained problem that combines *minimal deviation from the previous solution* with a *loss constraint*. Hereafter, unless otherwise specified, we focus exclusively on the inner iterative steps, which correspond to a single outer iterative step t.

Lagrangian form. Introducing a multiplier $\alpha \geq 0$ yields

$$\min_{\boldsymbol{W}} \max_{\alpha \geq 0} \mathcal{L}(\boldsymbol{W}, \alpha) = \frac{C}{p} \|\boldsymbol{W} - \boldsymbol{W}^{t-1}\|_p^p + \alpha\big(\phi(\boldsymbol{W}, \mathcal{X}^{(t)}) - \varepsilon\big). \tag{3}$$

and we alternate updates of $\boldsymbol{W}$ and α. When $\phi > \varepsilon$, α increases, strengthening the drive to reduce the data loss; after attainment, it trends back toward zero. Thus, α acts as an *on-demand gate* that intensifies learning only when needed.

3.2.3 Distance-Controlled Optimizer Design

We now detail the design of our optimizer, which incorporates distance control in a way that deliberately separates it from gradient statistics. The goal is to move from $\boldsymbol{W}^{t-1}$ to $\boldsymbol{W}$ with *only the change needed* to satisfy the current batch, thereby preserving prior knowledge.

Naive loss penalty vs. proximal correction. A straightforward approach adds the distance term directly to the data loss:

$$\underbrace{\alpha\phi(\boldsymbol{W}, \mathcal{X})}_{\text{data fit}} + \underbrace{\tfrac{C}{p} \|\boldsymbol{W} - \boldsymbol{W}^{t-1}\|_p^p}_{\text{distance control}}. \tag{4}$$

However, this contaminates the moment estimates in Adam-like optimizers. Instead, following the idea of RecAdam (Chen et al., 2020), we place distance control *outside* the loss and split the update into two stages. While RecAdam anchors parameters to the pretrained weights, we adapt this idea to our batch-wise setting by anchoring each update to the parameters from the previous batch, thereby enabling "stepwise RecAdam" updates.

1. **Data-fit update:** apply a standard optimizer, such as AdamW, to the scaled loss $\alpha\,\phi(\boldsymbol{W}, \mathcal{X})$ to obtain tentative weights $\boldsymbol{W'}$.
2. **Proximal correction:** project $\boldsymbol{W'}$ by the proximal operator of $g(\boldsymbol{W}) = \frac{C}{p}\|\boldsymbol{W} - \boldsymbol{W}^{(t-1)}\|_p^p$ to obtain the updated parameters.

This separation of roles keeps moment estimates purely data-driven, while distance control acts as a geometric post-update correction.

Scaling and gating of the data loss with α. To enforce the constraint $\phi(\boldsymbol{W}, \mathcal{X}) \leq \varepsilon$, we update the Lagrange multiplier α via projected gradient ascent. While a standard update would use a learning

rate $\eta^{(i)}$, our implementation adopts a more robust approach by fixing the step size and clipping the update magnitude. This prevents numerical instability when the loss ϕ is far from the target ε. The update is given by:

$$\alpha^{(i)} = \max(0, \alpha') \tag{5}$$

$$\alpha' = \alpha^{(i-1)} + \eta^{(i)}(\phi(\boldsymbol{W}^{(i)}, \mathcal{X}) - \varepsilon) \tag{6}$$

When $\phi > \varepsilon$, α grows, amplifying the pressure to fit the data; after attainment, it relaxes toward zero. The clipping mechanism ensures that this growth is controlled. Thus, α acts as a gate that increases drive only when necessary, helping to avoid overly aggressive updates.

Two-stage update: Loss minimization and proximal projection. Let $\texttt{Optim}(\cdot)$ denote an optimizer. We can apply any optimizer; however, for example, we often choose the Adam optimizer when training LMs. We assume $\texttt{Optim}(\cdot)$ to be such an optimizer. Then, each inner iteration performs

$$\boldsymbol{W'} = \texttt{Optim}\big(\alpha^{(i)}\,\phi(\boldsymbol{W}^{(i)}, \mathcal{X})\big), \tag{7}$$

$$\begin{aligned}\boldsymbol{W}^{(i+1)} &= \mathrm{prox}_{\eta g}(\boldsymbol{W'}) \\ &= \arg\min_{\boldsymbol{W}}\Big\{\eta\, g(\boldsymbol{W}) + \tfrac{1}{2}\,\|\boldsymbol{W} - \boldsymbol{W'}\|_2^2\Big\}.\end{aligned} \tag{8}$$

Closed forms follow from g. For $p=2$,

$$\boldsymbol{W}^{(i+1)} \;=\; \frac{1}{1+C}\,\big(\boldsymbol{W}^{t-1} + \boldsymbol{W'}\big). \tag{9}$$

i.e., a convex combination of $\boldsymbol{W'}$ and $\boldsymbol{W}^{t-1}$, with larger C pulling more strongly toward $\boldsymbol{W}^{t-1}$. For $p = 1$, the update reduces to elementwise *soft-thresholding*,

$$\boldsymbol{W}^{(i+1)} = \begin{cases} \boldsymbol{W'} - C, & \text{if } \boldsymbol{W}^{t-1} + C < \boldsymbol{W'}. \\ \boldsymbol{W}^{t-1}, & \text{if } |\boldsymbol{W'} - \boldsymbol{W}^{t-1}| \le C. \\ \boldsymbol{W'} + C, & \text{if } \boldsymbol{W'} < \boldsymbol{W}^{t-1} - C. \end{cases} \tag{10}$$

creating a width-C *dead band* around $\boldsymbol{W}^{t-1}$ that freezes small changes.

3.2.4 Algorithmic Summary of the Training Loop

At each batch $\mathcal{X}^{(t)}$, training begins with $\alpha = 1$ and repeatedly performs inner updates. Each step computes the cross-entropy loss $\phi(\boldsymbol{W}, \mathcal{X}^{(t)})$, updates the multiplier α by a *capped* projected ascent $\alpha \leftarrow \max\{0,\ \alpha + \min(1.0,\ \eta(\phi - \varepsilon))\}$, and then applies a two-stage parameter update: (i) an AdamW step on the scaled loss $\alpha\,\phi$, and (ii) a proximal correction with respect to $\boldsymbol{W}^{(t-1)}$. The loop exits once $\alpha < 1$ and shows no increase for three consecutive steps (or when $I_{\max}$ is reached), after which $\boldsymbol{W}^{(t)}$ is committed and training advances to the next batch.

Algorithm 1: Training with Alternating Lagrangian + Proximal Update

Input: batches $\{\mathcal{X}^{(t)}\}$, init. weights $\boldsymbol{W}^{(0)}$, distance coeff. C, norm p, threshold ε , learning rate η **for** $t = 1, 2, \ldots, T$ **do**

- Initialize $\alpha \leftarrow 1$, $\boldsymbol{W} \leftarrow \boldsymbol{W}^{(t-1)}$, no_inc $\leftarrow 0$, $i \leftarrow 0$;
- **repeat**
  - (1) $\phi \leftarrow \phi(\boldsymbol{W}, \mathcal{X}^{(t)})$; $\quad i \leftarrow i + 1$;
  - (2) $\Delta \leftarrow \min\big(1.0,\ \eta(\phi - \varepsilon)\big)$; $\alpha_{\text{new}} \leftarrow \max\{0,\ \alpha + \Delta\}$; **if** $\Delta \le 0$ **then** no_inc$\leftarrow$no_inc$+1$ **else** no_inc$\leftarrow 0$;
  - (3) $\boldsymbol{W'} \leftarrow \texttt{Optim}\big(\alpha_{\text{new}} \cdot \phi\big)$;
  - (4) **if** $p = 2$ **then** $\boldsymbol{W} \leftarrow \dfrac{\boldsymbol{W'} + C\,\boldsymbol{W}^{(t-1)}}{1 + C}$
  - **else if** $p = 1$ **then** $\boldsymbol{W} \leftarrow$ soft-threshold$(\boldsymbol{W'},\ \boldsymbol{W}^{(t-1)},\ C)$
  - (5) $\alpha \leftarrow \alpha_{\text{new}}$;
- **until** *($\alpha < 1$ **and** no_inc$= 3$)* **or** $i = I_{\max}$;
- Commit $\boldsymbol{W}^{(t)} \leftarrow \boldsymbol{W}$;

3.3 Parameter Averaging for Consolidation

Motivation. While our batch-wise constrained updates promote "mastery before progression," the sequence of batch endpoints $\{\overline{\boldsymbol{W}}^{(t)}\}_{t=1}^{T}$ can still exhibit oscillations as the model adapts to new material. Averaging successive solutions smooths these fluctuations, approximates an implicit ensemble, and biases the solution toward flatter minima, thereby reducing susceptibility to forgetting when exposure shifts.

Formulation. We adopt simple *arithmetic averaging* of multiple checkpoints:

$$\overline{\boldsymbol{W}} \;=\; \frac{1}{K}\sum_{j=1}^{K} \boldsymbol{W}^{(t_j)}, \tag{11}$$

which directly yields a consolidated parameter set without introducing additional hyperparameters. This averaged model serves as a geometry-based regularizer that complements the explicit distance

control, further reducing the effective drift across training.

3.4 Implementation Details

This subsection summarizes practical implementation details.

Optimizer re-initialization. Each batch is treated as an independent constrained optimization problem. Accordingly, we re-initialize the first and second moment states of the first-order optimizer at the beginning of every batch (instantiated as AdamW in our implementation). The global learning-rate schedule for model parameters is maintained consistently across training, while the Lagrange multiplier scales the loss side. This preserves a separation of roles: the optimizer explores each batch afresh, while the proximal correction preserves knowledge across batches.

Multiplier update and stopping rule. At the beginning of each batch we initialize $\alpha = 1.0$. The multiplier is updated via projected gradient ascent,

$$\alpha \leftarrow \Pi_{\alpha \geq 0}\big[\alpha + \eta\,(\phi - \varepsilon)\big],$$

with a fixed step size and clipping for numerical stability; concretely we clip the per-step increment by $\min(1.0,\ \eta(\phi - \varepsilon))$. For stopping, we monitor α rather than ϕ: convergence is declared once (i) $\alpha < 1$ and (ii) α does not increase for $M{=}3$ consecutive checks, corresponding to a *stable* satisfaction of $\phi \leq \varepsilon$. As a safeguard, we cap the number of inner iterations at $I_{\max}$.

Norm and penalty coefficient. We instantiate the distance term with an L_p-norm and, in our experiments, use $p \in \{1, 2\}$ with coefficient C. For the conceptual effect of each choice (e.g., contraction vs. dead-band behavior), see Section 3.2 (*Role of the distance term*).

Hyperparameters. The main hyperparameters are the distance coefficient C, norm p, threshold ε, inner learning-rate schedule $\{\eta^{(i)}\}$ for the multiplier update, iteration cap $I_{\max}$, and the required non-increase streak M. We instantiate ε from a target token accuracy via $\varepsilon = -\log a_{\text{tgt}}$ and set 0.6931 for $a_{\text{tgt}}{=}0.5$ as an initial reference, adjusting slightly to account for label smoothing and vocabulary effects; beyond these systematic shifts, we do not tune ε empirically.

Averaging protocol. For final consolidation, we perform *arithmetic* checkpoint averaging. We save checkpoints (i) every 1M words from 1M to 9M, (ii) every 10M words from 10M to 100M, and (iii) every 100M words from 100M to 1000M, yielding 28 checkpoints in total. Their simple average defines the final weights $\overline{\boldsymbol{W}}$ used for evaluation. Throughout training, all inner-loop decisions (loss thresholding, α updates, proximal correction) operate on the live weights $\boldsymbol{W}$.

4 Experiments

This study was conducted under the *Strict* track setting of the BabyLM Challenge 2025, following the official evaluation tasks and pipeline.[2]

4.1 Evaluation Tasks

We evaluate the proposed model on the benchmarks and tasks suggested by the BabyLM challenge organizers to assess its language acquisition capabilities.footnoteSee Appendix B for more details of evaluation datasets.

4.2 Baselines

4.2.1 Baselines for Leaderboard Comparison

We use the official BabyLM pretrained models from both the Strict and Loose Tracks as baseline comparisons. Specifically, we set GPT-BERT (Charpentier and Samuel, 2024) and GPT-2 small (Radford et al., 2019) as baselines.

4.2.2 Baselines for Controlled Comparison

To evaluate the effectiveness of our training algorithm and curriculum design, we additionally constructed three *controlled baselines*. These models were trained with vanilla pre-training using the same dataset but under different data scheduling strategies:

Random. The model was trained for 10 epochs on the proposed dataset with sentences randomly shuffled across the entire training process.

Curriculum. The model was trained for 10 epochs on the proposed dataset while preserving the curriculum ordering of the data.

Curriculum-Repeat. The model was trained on the curriculum-ordered dataset, but with each batch repeated consecutively 10 times before moving on to the next batch.

[2]Details regarding the experimental setup, including tokenizer selection, model architecture, and training procedures, are provided in Appendix D.

Model	BLiMP	BLiMP-S	EWoK	GLUE	WUG-ADJ	Entity	Reading(SPR/ET)	WUG-PAST	COMPS	AoA	Avg.
GPT-BERT	80.5	73.0	52.4	70.9	41.2	39.9	3.0/8.7	27.1	59.7	22.3	43.5
GPT-2 small	74.9	63.3	51.7	54.7	50.2	31.5	3.2/7.9	7.3	56.2	5.5	36.9
Proposed(115M)	49.2	50.4	50.2	57.7	57.5	41.8	0.1/0.6	4.3	49.4	0.0	32.8
Proposed(372M)	47.8	50.5	50.5	57.7	56.2	41.2	0.0/0.5	8.5	50.6	0.3	33.1

Table 2: Main results on the BabyLM evaluation suite. We compare public baselines with our proposed models at two scales (115M and 372M model). The GPT-BERT and GPT-2 small results are directly copied from the model cards released by the BabyLM organizers. For each scale, we report the best-performing configuration selected based on the highest average score across tasks.

Model (115M)	BLiMP	BLiMP-S	EWoK	GLUE	WUG-ADJ	Entity	Reading(SPR/ET)	WUG-PAST	COMPS	AoA	Avg.
Random	64.3	57.5	50.3	57.7	26.0	18.9	0.0/3.1	15.6	53.8	0.1	31.6
Curriculum	56.6	49.6	49.6	57.7	48.1	38.1	0.6/3.7	-12.6	50.3	0.3	31.1
Curriculum-Repeat	54.1	48.2	49.8	57.7	40.6	41.3	0.8/2.4	-10.9	50.2	0.2	30.4
Proposed (p=1, C=10^{-6})	49.2	50.4	50.2	57.7	57.5	41.8	0.1/0.6	4.3	49.4	0.0	32.8
Proposed (p=2, C=1)	52.2	50.2	49.4	57.7	57.1	41.1	0.0/1.5	-2.6	50.2	0.1	32.5

Table 3: Comparison of the 115M model on BabyLM evaluation tasks between controlled baselines (Random, Curriculum, Curriculum-Repeat) and our proposed method ($p = 1, C = 10^{-6}$; $p = 2, C = 1$).

Model (372M)	BLiMP	BLiMP-S	EWoK	GLUE	WUG-ADJ	Entity	Reading(SPR/ET)	WUG-PAST	COMPS	AoA	Avg.
Random	63.5	54.1	51.5	57.7	56.7	29.9	0.3/4.1	1.4	54.1	0.1	33.9
Curriculum	54.4	48.9	50.0	57.7	70.5	28.1	0.2/3.0	-16.2	50.9	0.0	31.6
Curriculum-Repeat	52.8	48.2	49.8	57.7	49.3	41.1	0.7/2.4	6.2	50.0	0.2	32.6
Proposed (p=1, C=10^{-6})	47.8	50.5	50.5	57.7	56.2	41.2	0.0/0.5	8.5	50.6	0.3	33.1
Proposed (p=2, C=1)	52.2	47.9	50.1	57.7	60.5	40.3	0.0/1.2	2.2	50.1	0.3	33.0

Table 4: Comparison of the 372M model on BabyLM evaluation tasks between controlled baselines (Random, Curriculum, Curriculum-Repeat) and our proposed method ($p = 1, C = 10^{-6}$; $p = 2, C = 1$).

These controlled baselines allow us to disentangle the contributions of curriculum structure and repetition effects from the impact of our proposed training algorithm.

5 Results

5.1 Main Results: Comparison with Public Baselines

Table 2 summarizes the main results on the BabyLM evaluation suite. For each model size, we report the configuration with the highest average score across all tasks. Both the 115M and 372M models use the same setting of $(p, C) = (1, 10^{-6})$. Overall, the proposed models perform lower than the official BabyLM baselines, GPT-BERT and GPT-2 small. Their average scores (around 33) remain below those baselines (43.5 and 36.9). This indicates that the constrained optimization used in our training has not yet reached the linguistic competence achieved by the official pretrained systems. The largest gaps appear in grammar-oriented benchmarks such as BLiMP and BLiMP-S. This underperformance may occur because the batch-wise constraint favors local stability over global syntactic generalization. In contrast, the models achieve relatively higher scores on WUG-ADJ and Entity Tracking, indicating stronger learning of morphological agreement and referential consistency. In summary, the proposed method provides small but consistent advantages in morphology- and discourse-level tasks, while remaining weaker on grammar-centric evaluations compared with the official baselines.

5.2 Controlled Comparison of the Proposed Method

Tables 3 and 4 compare the proposed constrained optimization with three controlled baselines that differ only in data-ordering and repetition. This analysis isolates the effect of the training algorithm itself. Overall, the proposed models achieve slightly higher average scores than the controlled baselines, but the improvements are small. For the 115M model, the proposed settings reach 32.5–32.8 in average, compared to 30–31 for the baselines. A similar pattern holds for the 372M model, where

Model	Settings	BLiMP	BLiMP-S	EWoK	GLUE	WUG-ADJ	Entity	Reading(SPR/ET)	WUG-PAST	COMPS	AoA	Avg.
115M	p=1, Curriculum	49.2	50.4	50.2	57.7	57.5	41.8	0.1/0.6	4.3	49.4	0.0	32.8
115M	p=1, Random	53.4	48.4	49.6	57.7	62.2	41.0	0.7/1.9	-4.6	49.6	-0.4	32.7
$\lvert\Delta\rvert$		4.2↓	2.0↑	0.6↑	0.0–	4.7↓	0.8↑	0.6↓/1.3↓	8.9↑	0.2↓	0.4↑	0.1↑
115M	p=2, Curriculum	52.2	50.2	49.4	57.7	57.1	41.1	0.0/1.5	-2.6	50.2	0.1	32.5
115M	p=2, Random	51.2	50.7	50.3	57.7	57.9	41.0	0.0/1.1	5.5	50.2	0.1	33.2
$\lvert\Delta\rvert$		1.0↑	0.5↓	0.9↓	0.0–	0.8↓	0.1↑	0.0–/0.4↑	8.1↓	0.0–	0.0–	0.7↓
372M	p=1, Curriculum	47.8	50.5	50.5	57.7	56.2	41.2	0.0/0.5	8.5	50.6	0.3	33.1
372M	p=1, Random	52.8	50.6	50.0	57.7	60.7	41.6	0.1/1.2	4.2	49.7	-0.0	33.5
$\lvert\Delta\rvert$		5.0↓	0.1↓	0.5↑	0.0–	4.5↓	0.4↓	0.1↓/0.7↓	4.3↑	0.9↑	0.3↑	0.4↓
372M	p=2, Curriculum	52.2	47.9	50.1	57.7	60.5	40.3	0.0/1.2	2.2	50.1	0.3	33.0
372M	p=2, Random	54.3	49.5	49.9	57.7	59.5	40.6	0.8/3.2	0.3	50.1	-0.3	33.2
$\lvert\Delta\rvert$		2.1↓	1.6↓	0.2↑	0.0–	1.0↑	0.3↓	0.8↓/2.0↓	1.9↑	0.0–	0.6↑	0.2↓

Table 5: Ablation study of curriculum scheduling for the 115M and 372M models. We compare $p = 1$ and $p = 2$ configurations trained on curriculum-ordered versus randomly shuffled data under the same C. Each Curriculum/Random pair is followed by a $|\Delta|$ row, reporting the absolute difference with ↑/↓ indicating whether curriculum ordering increased or decreased the score, respectively.

the proposed method yields around 33.0 on average, within the variance of baseline performance. These results suggest that the algorithm contributes incremental rather than substantial gains. In grammar-oriented benchmarks such as BLiMP, the proposed models tend to underperform, indicating that the constrained updates may restrict flexibility needed for broad syntactic learning. However, they show higher scores on WUG-ADJ and Entity Tracking, implying better capture of morphological and referential regularities. In summary, the proposed optimization provides minor yet consistent advantages in morphology- and discourse-level tasks, while overall performance remains comparable to the controlled baselines.

5.3 Effect of Curriculum Scheduling in the Proposed Method

Table 5 compares curriculum-ordered training with fully shuffled training data. For each model size, we report results under two representative hyperparameter settings, $(p, C) = (1, 10^{-6})$ and $(2, 1)$. Overall, the effect of curriculum scheduling is limited. Across both 115M and 372M models, the average differences between curriculum and random orders remain within one point, suggesting that the data order alone does not substantially influence final performance. In some grammar-related benchmarks such as BLiMP, the curriculum models even underperform, implying that gradual data presentation may constrain the diversity of contexts required for broader syntactic generalization. On the other hand, curriculum training yields slightly higher scores on WUG-ADJ and COMPS, indicating that staged exposure can help the model capture morphological and compositional patterns more consistently. In summary, curriculum ordering provides small and task-specific benefits but does not consistently improve overall language acquisition performance within the proposed framework.

6 Conclusion and Future Works

This study introduced a batch-wise constrained optimization framework for language model pretraining under the developmental data condition of the BabyLM Challenge 2025. Experimental results show that the method performs worse on grammar-sensitive benchmarks such as BLiMP and BLiMP-S, suggesting that the constraint limits generalization across syntactic contexts. In contrast, it yields small but consistent gains on morphology- and discourse-related tasks such as WUG-ADJ and Entity Tracking, indicating slightly improved consistency in local linguistic patterns.

Future work will address the theoretical gap between batch-wise learning and the distributional hypothesis underlying language modeling. Learning from isolated batches restricts contextual diversity, which likely explains the weak syntactic generalization observed. A promising direction is to extend the framework toward context-aggregated or memory-based architectures that integrate information across batches.

Acknowledgments

This work was supported by the "R&D Hub Aimed at Ensuring Transparency and Reliability of Generative AI Models" project of the Ministry of Education, Culture, Sports, Science and Technology, and JST Moonshot R&D Grant Number JPMJMS2011-35 (fundamental research). In this study, we mainly used ABCI 3.0 and the computer resource offered by Research Institute for Information Technology, Kyushu University under the category of General Projects. ABCI 3.0 is provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use".

References

Rufus Behr. 2024. ELC-ParserBERT: Low-resource language modeling utilizing a parser network with ELC-BERT. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 140–146, Miami, FL, USA. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Arielle Borovsky, Jeffrey L Elman, and Anne Fernald. 2012. Knowing a lot for one's age: Vocabulary skill and not age is associated with anticipatory incremental sentence interpretation in children and adults. *Journal of experimental child psychology*, 112(4):417–436.

Luca Capone, Alessandro Bondielli, and Alessandro Lenci. 2024. ConcreteGPT: A baby GPT-2 based on lexical concreteness and curriculum learning. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 189–196, Miami, FL, USA. Association for Computational Linguistics.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, Online. Association for Computational Linguistics.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [Call for Papers] The 2nd BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2404.06214*.

Gina Conti-Ramsden and Kevin Durkin. 2012. Language development and assessment in the preschool period. *Neuropsychology Review*, 22(4):384–401.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Emmanuel Dupoux. 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

Steven Y. Feng, Noah D. Goodman, and Michael C. Frank. 2024. Is child-directed speech effective training data for language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 22055–22071, Miami, Florida, USA. Association for Computational Linguistics.

Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135.

Mohammad Amin Ghanizadeh and Mohammad Javad Dousti. 2024. Towards data-efficient language models: A child-inspired approach to language learning. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 22–27, Miami, FL, USA. Association for Computational Linguistics.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, D. Kimbrough Oller, John H. L. Hansen, and Terrance D. Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

Akari Haga, Akiyo Fukatsu, Miyu Oba, Arianna Bisazza, and Yohei Oseki. 2024. BabyLM challenge: Exploring the effect of variation sets on language model training efficiency. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 252–261, Miami, FL, USA. Association for Computational Linguistics.

Valentin Hofmann, Leonie Weissweiler, David Mortensen, Hinrich Schütze, and Janet Pierrehumbert. 2024. Derivational morphology reveals analogical generalization in large language models. *Preprint*, arXiv:2411.07990.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *Preprint*, arXiv:2405.09605.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44(4):978–990.

David Lopez-Paz and Marc' Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Brian Macwhinney. 2000. The childes project: tools for analyzing talk. *Child Language Teaching and Therapy*, 8.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Mir Tafseer Nayeem and Davood Rafiei. 2024. KidLM: Advancing language models for children – early insights and future directions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4813–4836, Miami, Florida, USA. Association for Computational Linguistics.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Preprint*, arXiv:1901.04513.

Tracy Reuter, Arielle Borovsky, and Casey Lew-Williams. 2019. Predict and redirect: Prediction errors support children's word learning. *Developmental psychology*, 55(8):1656.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2022. Progressive neural networks. *Preprint*, arXiv:1606.04671.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is more: Pre-training cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 174–188, Miami, FL, USA. Association for Computational Linguistics.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. *Preprint*, arXiv:1911.11763.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. {LAMAL}: {LA}nguage modeling is all you need for lifelong language learning. In *International Conference on Learning Representations*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings*

of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schütze, Kemal Oflazer, and David R. Mortensen. 2023. Counting the bugs in chatgpt's wugs: A multilingual investigation into the morphological capabilities of a large language model. *Preprint*, arXiv:2310.15113.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1112–1125, Online. Association for Computational Linguistics.

A Overview of Prior Approaches to Catastrophic Forgetting

Replay-based methods. In natural language processing, representative replay-based methods include Gradient Episodic Memory (GEM) (Lopez-Paz and Ranzato, 2017) and LAnguage MOdeling for Lifelong language learning (LAMOL) (Sun et al., 2020). Both methods aim to preserve performance on previously learned tasks when learning a new one. GEM explicitly stores examples from past tasks and constrains gradient updates so as not to reduce past task performance, whereas LAMOL avoids explicit memory by generating pseudo-samples from past tasks and mixing them with new training examples.

Regularization-based methods. Regularization is typically employed to prevent overfitting, but it has also been adapted to mitigate forgetting. Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) and Synaptic Intelligence (SI) (Zenke et al., 2017) slow down the update of parameters deemed important for previously learned tasks, thereby maintaining prior knowledge while enabling new learning.

Architectural methods. Approaches such as Progressive Neural Networks (PNNs) (Rusu et al., 2022) expand the network by adding new parameters for each task. This enables the reuse of knowledge from past tasks while preventing forgetting. However, these methods require identifying which parameters correspond to each task, and the overall model size grows linearly with the number of tasks, which poses practical challenges.

Why regularization in this study? In the BabyLM setting, where the vocabulary size and available training data are restricted, replay-based methods that rely on storing or generating additional samples are less desirable. Similarly, parameter expansion approaches are impractical for complex language model architectures. For these reasons, this study adopts a regularization-based approach to mitigate forgetting.

B Details of Evaluation Tasks

BLiMP. BLiMP (Benchmark of Linguistic Minimal Pairs) (Warstadt et al., 2020) is an English zero-shot benchmark that tests the grammatical knowledge of language models. We also use BLiMP Supplement, an extended version covering dialogue, question–answer congruence, and lexical semantics.

EWoK. Ewok (Elements of World Knowledge) (Ivanova et al., 2025) assesses the extent to which LMs possess fundamental world knowledge. It uses a cognition-inspired framework to evaluate whether models can identify plausible contexts given varying fillers.

GLUE. GLUE (Wang et al., 2018) is a multi-task benchmark for evaluating natural language understanding models. As models began surpassing human performance on GLUE, the more challenging SuperGLUE (Sarlin et al., 2020) was introduced to provide a tougher evaluation.

Derivational Morphology. This task tests a language model's ability to generate English adjective-to-noun nominalizations (Hofmann et al., 2024), focusing on irregular patterns that defy simple rules.

Entity Tracking. This task measures how well LMs can track changes in the states of entities throughout a text (Kim and Schuster, 2023). Given a description of an entity's initial state and a sequence of state-altering events, the model must infer the entity's final state.

Reading time. This task assesses how well an LM's word prediction probabilities align with human cognitive processing data (de Varda et al., 2024). The task examines the relationship between model predictions and brain responses or reading behaviors.

WUG PAST. This hidden task evaluates morphological generalization by correlating model and human distributions for nonce verb past-tense and adjective nominalizations (Weissweiler et al., 2023; Hofmann et al., 2024), enabling comparison across evaluation protocols.

COMPS. This task tests whether models correctly inherit properties from superordinate to subordinate concepts using nonce-word minimal pairs (Misra et al., 2023). Models should assign higher probability to the semantically correct sentence, probing robust conceptual inheritance in language representations.

Age of Acquisition. This task estimates when models learn specific words by tracking surprisal across training (Chang and Bergen, 2022). Model-derived acquisition ages are fit with sigmoids and compared to human norms from the

Component	115M Model	372M Model
Total parameters	114,964,224	372,234,112
Number of layers	12	24
Hidden size	768	896
Attention heads	8	14
FFN intermediate size	2,688	4,864
Activation	SiLU	SiLU
Normalization	RMSNorm ($\epsilon = 10^{-6}$)	RMSNorm ($\epsilon = 10^{-6}$)
Positional encoding	RoPE ($\theta = 10^{6}$)	RoPE ($\theta = 10^{6}$)
Vocab size	16k	16k

Table 6: Architectural specifications of the 115M and 372M models. Both follow the Qwen2 decoder-only architecture with differences in scale.

MacArthur–Bates Communicative Development Inventory.

C Difficulty scoring for curriculum ordering

To rank samples, we combine five textual features:

- L_i: average sentence length (words per sentence)
- $TTR_i^{(1/2)}$: square-root adjusted type–token ratio
- R_i: rare-word rate (fraction not found in a background frequency lexicon)
- AoA_i^{mean}: mean Age of Acquisition over tokens
- $AoA_i^{\max}$: maximum Age of Acquisition over tokens

Because these features have different scales, we standardize each x_i by a z-score:

$$z(x_i) = \frac{x_i - \mu_x}{\sigma_x}. \tag{12}$$

Here, we define $\mathbf{f}_i$ and $\mathbf{w}$ as follows:

$$\mathbf{f}_i = \left(L_i,\ TTR_i^{(1/2)},\ R_i,\ AoA_i^{\text{mean}},\ AoA_i^{\max}\right),$$
$$\mathbf{w} = (0.20,\ 0.15,\ 0.20,\ 0.20,\ 0.05).$$

We define the composite difficulty score as:

$$\text{Score}_i = \sum_{k=1}^{5} w_k \cdot z(f_{k,i}), \tag{13}$$

and sort items in ascending order of Score_i. The weights prioritize structural length and lexical rarity while incorporating average and worst-case AoA to align with developmental plausibility.

D Experimental Setup

Tokenizer. In this study, we trained a new tokenizer directly on the pre-training dataset. To ensure sufficient expressiveness under the limited-data setting, we adopted the Unigram subword segmentation model. The Unigram model is known to provide flexible segmentation for infrequent and morphologically varied words, thereby balancing vocabulary compression with generalization ability (Kudo and Richardson, 2018). Concretely, we trained the tokenizer using the Hugging Face `tokenizers` implementation, with the vocabulary size fixed at 16,000. We additionally included the special tokens <pad>, <cls>, <sep>, <s>, </s>, and <unk>. For text normalization, we applied the NFKC scheme, and we enabled byte fallback to guarantee stable encoding even for inputs containing unseen characters.

Model Architecture. We adopted a Transformer-based decoder-only language model, motivated by findings that children engage in predictive sentence processing: they integrate syntactic and semantic cues to anticipate upcoming words (Borovsky et al., 2012), which in turn facilitates sentence structure learning (Reuter et al., 2019). For similar reasons, decoder-only architectures are widely used in BabyLM challenge (Haga et al., 2024; Warstadt et al., 2023; Hu et al., 2024).

Table 6 lists the architectural specifications of the two model scales used in our experiments: 115M and 372M model. Both models follow the Qwen2 (Yang et al., 2024) architecture family, sharing the same decoder-only backbone but differing in size-related hyperparameters.

Pre-training parameters. Table 7 lists the parameters that are common to both the baselines and the proposed method, covering general optimization and compute settings. Table 8 specifies the hy-

perparameters unique to our proposed method, as well as those that differ from the baselines. All experiments were trained on a single NVIDIA H200 (64GB) GPU, with a maximum wall-clock training time of approximately 24 hours per run for the proposed models.

Training hyperparameter	Setting
Total training tokens	1,176,922,330
Batch size	128
Sequence length	512
Warmup ratio	5%
Learning rate (max/min)	5e-4 / 5e-5
Scheduler	cosine
Optimizer	AdamW
AdamW $\beta_1, \beta_2, \epsilon$	0.9, 0.999, 1e-8
GPU hardware	1 $\times$ NVIDIA H200 (64GB)

Table 7: Common training hyperparameters and compute setup shared across all baselines and proposed models.

Hyperparameter	Baselines	Proposed
AdamW weight decay	0.01	0
Optimizer re-initialization	–	Enabled
p (norm type)	–	1 , 2
C (penalty coefficient)	–	10^{-6} , 1
Cross-entropy tolerance ϵ	–	0.693 ($-\ln 0.5$)
η for α update	–	0.1
max batch repeat	–	10

Table 8: Hyperparameters specific to the proposed method and those differing from the baselines.

E Additional Ablation: Optimizer Re-initialization

Table 9 presents an ablation of optimizer state re-initialization. Overall, removing re-initialization (w/o) does not consistently improve performance and sometimes leads to instability across tasks. For instance, the 115M and 372M models without re-initialization show large fluctuations in WUG-PAST and BLiMP scores, suggesting that momentum carried over between batches can introduce noise and interfere with the batch-level independence assumed by our method. At the same time, re-initialization slightly decreases scores on a few morphology-oriented benchmarks such as WUG-ADJ, indicating that the added stability may also reduce optimization flexibility. Across all settings, the differences in average scores remain within one point, confirming that the choice has only minor quantitative impact. In summary, re-initializing the optimizer state helps maintain the intended separation between batches but provides limited benefit in terms of overall downstream performance.

Model	p	Re-initialization	BLiMP	BLiMP-S	EWoK	GLUE	WUG-ADJ	Entity	Reading(SPR/ET)	WUG-PAST	COMPS	AoA	Avg.
115M	1	w/	49.2	50.4	50.2	57.7	57.5	41.8	0.1/0.6	4.3	49.4	0.0	32.8
115M	1	w/o	55.2	47.7	50.0	57.7	60.8	40.9	0.7/4.1	-13.1	50.3	0.0	32.2
$\lvert\Delta\rvert$			6.0↓	2.7↑	0.2↑	0.0–	3.3↓	0.9↑	0.6↓/3.5↓	17.4↑	0.9↓	0.0–	0.6↑
115M	2	w/	52.2	50.2	49.4	57.7	57.1	41.1	0.0/1.5	-2.6	50.2	0.1	32.5
115M	2	w/o	53.8	50.8	49.9	57.7	48.6	40.9	0.3/2.2	5.2	49.6	0.1	32.6
$\lvert\Delta\rvert$			1.6↓	0.6↓	0.5↓	0.0–	8.5↑	0.2↑	0.3↓/0.7↓	7.8↓	0.6↑	0.0–	0.1↓
372M	1	w/	47.8	50.5	50.5	57.7	56.2	41.2	0.0/0.5	8.5	50.6	0.3	33.1
372M	1	w/o	56.5	49.6	50.5	57.7	60.5	39.9	1.9/3.6	-1.0	49.8	0.1	33.5
$\lvert\Delta\rvert$			8.7↓	0.9↑	0.0–	0.0–	4.3↓	1.3↑	1.9↓/3.1↓	9.5↑	0.8↑	0.2↑	0.4↑
372M	2	w/	52.2	47.9	50.1	57.7	60.5	40.3	0.0/1.2	2.2	50.1	0.3	33.0
372M	2	w/o	52.8	46.9	50.3	57.7	57.8	41.2	0.2/2.0	-4.6	49.5	0.8	32.2
$\lvert\Delta\rvert$			0.6↓	1.0↓	0.2↑	0.0–	2.7↓	0.9↓	0.2↓/0.8↓	6.8↑	0.6↑	0.5↓	0.8↑

Table 9: Ablation study on optimizer state re-initialization for the 115M and 372M models. Each pair of rows compares training with (w/) and without (w/o) re-initializing optimizer states at the beginning of each batch. Following each pair, the $|\Delta|$ row reports the absolute difference with ↑/↓ indicating whether re-initialization increased or decreased the score, respectively.

Pretraining Language Models with LoRA and Artificial Languages

Nalin Kumar and Mateusz Lango and Ondřej Dušek
Charles University, Faculty of Mathematics and Physics, Prague, Czechia
{nkumar,lango,odusek}@ufal.mff.cuni.cz

Abstract

Large language models (LLMs) require a substantial amount of training data, which contrasts with the data-efficient learning observed in humans. In our submission to the BabyLM Challenge, we address this disparity by proposing a parameter-efficient *pretraining* approach for language acquisition from limited data. Our approach involves initializing the model with token embeddings trained by a shallow model, followed by tuning the non-embedding parameters with non-linguistic data to introduce structural biases. Then, we freeze the resulting model and pretrain it on the 10M-token BabyLM corpus using LoRA adapters. Experiments on small corpora demonstrate that our approach improves upon classic pretraining of the entire model.

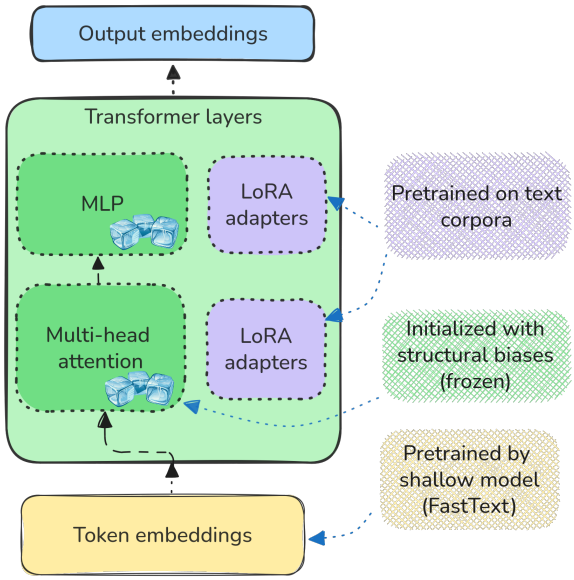


Figure 1: Overview of the proposed parameter-efficient pretraining.

1 Introduction

Large language models (LLMs) have shown impressive performance across a wide range of benchmarks, often rivaling human capabilities. However, their training requires far more data than humans need to acquire knowledge. To address the gap between the training efficiency of LLMs and that of a child, the BabyLM challenge provides an evaluation framework for developing data-efficient language models trained on human-scale training data of 10M–100M words (Warstadt et al., 2023; Hu et al., 2024; Charpentier et al., 2025).

This paper presents our submission to the strict-small track of the BabyLM Challenge, which aims to train high-performance language models using a corpus of just 10 million words. Our work focuses on developing parameter-efficient architectures for model pretraining, as smaller models typically achieve better results when trained on small datasets by reducing the risk of overfitting.

The proposed model is based on a BERT-like transformer architecture (Devlin et al., 2019), randomly initialized and pretrained on non-linguistic data (correct bracketing) designed to inject language-inspired structural biases into the model. After the initial pretraining stage, the entire model is frozen and further training on human-written texts is carried out using low-rank adapters (LoRA, Hu et al., 2021). This significantly reduces the number of parameters trained on text corpora compared to the original model. To further enhance parameter efficiency during pretraining, we also explore the use of shallow models for word embedding construction to initialize the embedding matrices.

An experimental evaluation on two popular benchmarks, BLiMP (Warstadt et al., 2020) and EWoK (Ivanova et al., 2024), reveals that models trained with the proposed parameter-efficient pretraining outperform those trained with standard pretraining of all parameters. Ablation experiments further demonstrate the benefits of non-linguistic

Proceedings of the First BabyLM Workshop, pages 537–542

data initialization and shallow models for embedding matrices. We publish our results and submitted models on HF repository.

2 Related Works

Word embeddings Early work demonstrating that models pretrained on free-text corpora can be useful for knowledge transfer across multiple deep learning tasks, was primarily focused on constructing word embeddings. Mikolov et al. (2013) proposed word2vec, which learns word embeddings using a skip-gram objective. Subsequently, the mathematical relation between word2vec and matrix decomposition was exploited to propose GloVe (Pennington et al., 2014). Later, the word2vec framework was extended to FastText (Bojanowski et al., 2017) that represents words as sums of character n-gram embeddings, with a hashing trick applied to improve parameter efficiency. All of these models are shallow and very fast to train, even on CPUs, yet they may provide valuable initialization points for embedding matrices in neural models (Kim, 2014). Note that embeddings constitute a significant fraction of the parameters of smaller models; for instance, in BERT (Devlin et al., 2019), roughly one-fifth of all parameters are devoted to token representations, so their good initialization may provide significant performance benefits.

Pretraining on artificial languages The effectiveness of pre-trained language models in performing downstream tasks has sparked considerable research interest in understanding the underlying reasons. This was often investigated using specially designed artificial languages. Papadimitriou and Jurafsky (2020) noticed that pretraining on non-linguistic data, such as MIDI music or sequences of pairs of matched integers, enhances the performance of language models on downstream tasks. Chiang and yi Lee (2022) further studied pretraining on integer strings, measuring its influence on the results on GLUE benchmark. They found that training on integer strings with the same unigram or bigram distributions as English words had a minimal effect on fine-tuning. Conversely, training on strings with stronger dependencies, e.g., containing groups of shuffled consecutive numbers or sequences of paired integers, resulted in significant improvements. These observations were also confirmed by Ri and Tsuruoka (2022). The injection of structural biases into models via pretraining on specific artificial languages was studied by Papadimitriou and Jurafsky (2023), whose experiments showed a reduction in perplexity of up to four times compared to random initialization.

Parameter-efficient fine-tuning The aim of parameter-efficient fine-tuning (PEFT) methods is to adapt large language models (LLMs) for downstream tasks by updating only a small subset of parameters, significantly reducing the computational and memory requirements. Techniques such as adapters (Chronopoulou et al., 2023), prefix tuning (Li and Liang, 2021), or fine-tuning only bias terms (Ben Zaken et al., 2022) have demonstrated competitive performance with full fine-tuning. A popular PEFT method is LoRA (Hu et al., 2021), which freezes the pretrained parameters of language models and approximates updates to weight matrices by a low-rank decomposition. To the best of our knowledge, such techniques were not previously used for language model *pretraining*.

3 Proposed Methodology

In this work, we propose a three-step approach for parameter-efficient pretraining on small text corpora: (1) using shallow model embeddings for better surface-level lexical representation, (2) initializing the language model with structural biases by pretraining it on artificial languages, and (3) pretraining the frozen model using LoRA adapters.

3.1 Initialization with pretrained embeddings

As token embedding matrices constitute a significant fraction of the parameters of small language models, we initialize them by optimizing the continuous skip-gram model objective (Mikolov et al., 2013) with a shallow linear model implemented in the FastText package (Bojanowski et al., 2017).

The word embedding model is trained using the same text corpus as the full language model, but with additional preprocessing. As FastText provides word embeddings and neural language models operate on tokens, the entire corpus is tokenised with a whitespace character introduced to split words into tokens. Next, standard FastText training is performed, with the word embedding size set to that of the neural model's input embeddings. The embedding layer of the language model is then initialised by the pretrained FastText embeddings.

3.2 Initialization with artificial languages

To initialize the model with structural biases, we experiment with pretraining on two artificial languages proposed by Papadimitriou and Jurafsky (2023): the nested parentheses language (NEST) and the crossing parentheses language (CROSS).

The NEST language has a vocabulary containing pairs of opening and closing tokens. Text is generated from left to right, with an opening token chosen with probability $p = 0.49$ and a closing token chosen with probability $p = 0.51$ in each iteration. If a closing token is selected, the most recently unmatched opening token is closed. An example sentence from the NEST language is:

$$1_(\, 24_(\, 24_) \, 67_(\, 39_(\, 39_) \, 67_) \, 1_)$$

The CROSS language operates on the same vocabulary as NEST; the difference is that the closing token can appear in any position after the opening token. Therefore, every NEST sequence is a correct CROSS sequence, but not vice versa, e.g.:

$$1_(\, 24_(\, 67_(\, 24_) \, 39_(\, 39_) \, 1_) \, 67_)$$

is a correct CROSS sentence. The text generation procedure of CROSS keeps the distribution of distances between the opening and closing tokens the same as in the NEST language.

The corpora generated in these two languages are applied for the initial pretraining of the transformer model with the standard masked language modeling objective. In this way, we can teach the model structural biases present in (non-)context-free grammars without using any language data and enable more efficient training on a small dataset.

3.3 Parameter-efficient pretraining

After initializing the transformer language model with pretrained embeddings and structural biases (on artificial languages), we freeze the weights of the entire model and inject LoRA's trainable rank decomposition matrices into each layer. The model is then trained with a standard masked language modelling (MLM) objective with default parameters from HuggingFace library (Wolf et al., 2020).

4 Experimental setup

4.1 Dataset

For pretraining on text data, we use the 10M-words version of BabyLM Corpus (Charpentier et al., 2025), comprising data sampled from 6 different domains. It includes OpenSubtitles (20%; dialogue from films), Simple English Wikipedia (15%; nonfiction), BNC (8%; dialogue), Project Gutenberg (26%; fiction & nonfiction), CHILDES (29%; dialogue), and Switchboard (1%; dialogue).

For all our experiments, we use the cased variant of the pretrained BERT tokenizer with a vocabulary size of 28k. The non-linguistic pretraining data consisted of 20,000 integer sequences following the grammar of artificial languages. Each sequence contained 512 tokens with a vocabulary size of 28k.

4.2 Autmomatic Evaluation Metrics

Models submitted to the BabyLM strict-small track are evaluated using a suite of automatic evaluation metrics: BLiMP (Warstadt et al., 2020), EWoK (Ivanova et al., 2024), GLUE (Wang et al., 2019), Entity Tracking (Kim and Schuster, 2023), WUG Adjective Nominalization (Hofmann et al., 2025), WUG Past Tense (Weissweiler et al., 2023), COMPS (Misra et al., 2023), Reading Cloze (de Varda et al., 2024), and AoA (Chang and Bergen, 2022). In this work, we report our results only for BLiMP and EWoK benchmarks.

4.3 Training details

FastText embeddings of dimension 768 were trained using the gensim library.[1] The training employed the skip-gram objective with a window size of 5 and was optimized for 5 epochs.

Pretraining on artificial languages was performed with the default HuggingFace Trainer hyperparameters, namely the AdamW optimizer with a learning rate of $5 \cdot 10^{-5}$ and a dynamic batch size. The optimization was performed for 25 epochs using the masked language modeling objective with a masking probability of 0.20.

We experimented with ranks $16, 64, 128, 256$ of LoRA (Hu et al., 2021), always setting the parameter α to twice the rank (i.e., $\alpha = 2 \cdot \text{rank}$). LoRA adapters were applied to all modules except the input and output embeddings and trained for 10 epochs. The pretraining setup otherwise followed the same hyperparameters described above.

4.4 Model Variants

All experiments use the BERT-base architecture (Devlin et al., 2019) as the underlying language model. In addition to testing the proposed approach, we perform experiments on various ablations to assess the contribution of each component.

[1] https://pypi.org/project/gensim/

Model			BLiMP	Supp.	EWoK	Avg
Embedding init.	AL init.	Pretraining				
BERT-base (Devlin et al., 2019) skyline			84.15	69.84	55.75	69.91
Model initializations						
Random	None	None	54.91	47.25	50.09	50.75
FastText	NEST	None	52.25	49.13	50.04	50.47
FastText	CROSS	None	57.51	50.05	**50.47**	52.67
Pretrained models						
Random	None	Standard	56.26	48.48	50.09	51.61
Random	None	LoRA	53.09	46.25	49.97	49.77
Random	CROSS	LoRA	52.66	45.32	50.11	49.36
Random	CROSS	LoRA + emb.	54.14	45.68	49.74	49.85
FastText	NEST	LoRA	55.99	51.73	50.02	52.58
FastText	CROSS	LoRA	**58.18**	**51.98**	50.38	**53.51**

Table 1: Evaluation results of trained language models on the 10M corpus with different initializations of embedding matrices (Embedding init.), initial pretraining on artificial languages (AL init.) and pretraining methods. LoRA + emb. indicates fine-tuning of LORA adapters together with input and output embedding matrices. LoRA is tested with the default rank of 16. Scores are measured on BLiMP, BLiMP Supplement (Supp.) and EWok benchmarks, with the Avg column showing an average of all three values.

LoRA rank	BLiMP	Supp.	EWoK	Avg
16	58.18	51.98	50.38	53.51
64	58.55	50.49	**50.43**	53.15
128	**60.96**	51.27	50.25	54.16
256	60.20	**53.21**	50.10	**54.50**

Table 2: Results of our approach (initialized by FastText and CROSS language) with different ranks of LoRA matrices (see Table 1 for scores).

Model Initializations We evaluate model performance without pretraining on linguistic data. Specifically, we evaluate the performance of completely randomly initialized language model, as well as the models initialized by FastText and pretrained on CROSS and NEST artificial languages.

Pretrained Models We also investigate several variants of pretrained models. The primary baseline is a transformer model trained in the standard way: weights are randomly initialized, and pretraining updates all model parameters. We then test models with LoRA adapters and word embeddings initialized either randomly or with FastText. Similarly, variants initialized with the NEST and CROSS artificial languages are evaluated.

5 Results

Table 1 presents the automatic evaluation scores on BLiMP, BLiMP Supplement (Supp.), and EWoK. Among the compared settings, FastText-CROSS-LoRA achieves the best performance, showing a gain of approximately four points over its counterpart initialized with random embeddings (Random-CROSS-LoRA). Overall, initializing the model with FastText embeddings consistently outperforms random initialization. Artificial language pretraining appears beneficial only in the CROSS setting, while configurations using NEST tend to degrade performance. Interestingly, the model pretrained only on artificial languages with FastText initialization obtained better performance than the standard pretraining on text data. LoRA-based pretraining yields slightly better results on BLiMP and BLiMP Supplement benchmarks.

Since LoRA introduces only a small number of trainable parameters and the default rank of 16 is designed for fine-tuning only, the model may not have sufficient capacity for pre-training and thus underfit on the BabyLM corpus. To address this, we experimented with higher LoRA matrix ranks (see results in Table 2). For BLiMP, performance generally improves as the rank increases, but slightly drops at a higher value, 256. In the case of BLiMP Supp., the highest LoRA rank yields the best results. By contrast, similar to BLiMP Supp., performance on EWoK does not show any consistent correlation with increasing rank.

6 Summary

This paper presents a parameter-efficient approach for pretraining language models on small text corpora. The main innovations include the usage of artificial languages to induce structural biases, using shallow models for matrix embedding initialization and pretraining a large model with LoRA adapters.

Limitations

This paper was limited in testing different configurations of trained models and it is highly probable that the training parameters used were not optimal.

Acknowledgments

This work was supported by the European Research Council (Grant agreement No. 101039303, NG-NLG) and Grant Agency of Charles University (Grant No. 302425), and used resources of the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).

References

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Cheng-Han Chiang and Hung yi Lee. 2022. On the transferability of pre-trained language models: A study from artificial datasets. *Preprint*, arXiv:2109.03537.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023. AdapterSoup: Weight averaging to improve generalization of pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, Dubrovnik, Croatia. Association for Computational Linguistics.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Valentin Hofmann, Leonie Weissweiler, David R Mortensen, Hinrich Schütze, and Janet B Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, and 1 others. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Preprint*, arXiv:1408.5882.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Preprint*, arXiv:2101.00190.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Isabel Papadimitriou and Dan Jurafsky. 2020. Learning Music Helps You Read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839, Online. Association for Computational Linguistics.

Isabel Papadimitriou and Dan Jurafsky. 2023. Injecting structural hints: Using language models to study inductive biases in language learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8402–8413, Singapore. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ryokan Ri and Yoshimasa Tsuruoka. 2022. Pretraining with artificial language: Studying transferable knowledge in language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7302–7315, Dublin, Ireland. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Masked Diffusion Language Models with Frequency-Informed Training

Despoina Kosmopoulou[1,2] **Efthymios Georgiou**[3] **Vaggelis Dorovatas**[2]
Georgios Paraskevopoulos[4] **Alexandros Potamianos**[1,2]

[1] National Technical University of Athens
[2] Archimedes RU, Athena RC
[3] University of Bern
[4] Institute of Language and Signal Processing, Athena RC
despoinakkosmopoulou@gmail.com efthymios.georgiou@unibe.ch

Abstract

We present a masked diffusion language modeling framework for data-efficient training for the *BabyLM 2025 Challenge*. Our approach applies diffusion training objectives to language modeling under strict data constraints, incorporating frequency-informed masking that prioritizes learning from rare tokens while maintaining theoretical validity. We explore multiple noise scheduling strategies, including two-mode approaches, and investigate different noise weighting schemes within the Negative Evidence Lower Bound (NELBO) objective. We evaluate our method on the BabyLM benchmark suite, measuring linguistic competence, world knowledge, and human-likeness. Results show performance competitive to state-of-the-art hybrid autoregressive-masked baselines, demonstrating that diffusion-based training offers a viable alternative for data-restricted language learning.

1 Introduction

By the age of 12, human children are typically exposed to fewer than 100 million words (Gilkerson et al., 2017). In contrast, state-of-the-art language models (LMs) (Touvron et al., 2023; Qwen et al., 2025) are trained on trillions of tokens. The BabyLM Challenge (Warstadt et al., 2023a) was introduced to address this striking efficiency gap by encouraging research on more data-efficient pretraining strategies. The *2025 strict track* constrains participants to train models for up to 10 epochs on a 100M-word corpus (Charpentier et al., 2025).

A prominent recent approach, winning the 2024 iteration of the BabyLM Challenge, `GPT-BERT`, combined a Masked Language Modeling (MLM) and Next Token Prediction (NTP) objective during pretraining (Charpentier and Samuel, 2024). The MLM objective has limited learning (gradient signal) efficiency, utilizing only $\sim$15% of corpus tokens per epoch (Devlin et al., 2019), while NTP learns from all tokens; as a result, NTP-based autoregressive (AR) generative models dominate the landscape of state-of-the-art language modeling (Brown et al., 2020). However, AR models typically use causal attention, only attending to previous tokens, which limits their bidirectional understanding and expressive ability (Devlin et al., 2019).

Recent advances in diffusion models have enabled their application to discrete text generation, with masked diffusion language models (`MDLMs`) emerging as a promising approach that combines bidirectional context modeling with generative training (Sahoo et al., 2024). `MDLMs` are masked language models with "parallel" generative capabilities, offering a compelling middle ground between the bidirectional understanding of MLMs and the generative efficiency of AR models. Unlike traditional MLM where a fixed percentage of tokens is masked at each step, `MDLMs` employ a diffusion process that varies masking rates across training, potentially leading to more efficient learning dynamics. This creates a natural curriculum where the model learns to reconstruct text under varying levels of corruption.

Recent work has shown that `MDLMs` can achieve competitive performance with AR models, while maintaining the bidirectional context benefits of masked models (Sahoo et al., 2024; Shi et al., 2025). However, diffusion models face challenges in data-sparse settings, with their multi-step training process potentially amplifying overfitting issues, an area that remains relatively unexplored in language modeling. Specifically, `MDLMs`' effectiveness in data-constrained settings remains unknown. In this work, we explore whether `MDLMs` trained for just 10 epochs over a 100M word corpus can match or surpass hybrid state-of-the-art approaches like `GPT-BERT`.

We *hypothesize* that the principled diffusion training objective of `MDLMs`, combined with strate-

Proceedings of the First BabyLM Workshop, pages 543–551

gic masking approaches, can achieve more sample-efficient learning compared to fixed-rate MLM or purely autoregressive training. To test this hypothesis, we implement a masked diffusion language modeling framework and explore multiple noise scheduling strategies, including two-mode approaches, while investigating different noise weighting schemes within the Negative Evidence Lower Bound (NELBO) objective. We further introduce frequency-informed masking that progressively prioritizes learning from rare tokens during the diffusion process, directing the model's attention toward more informative and challenging aspects of language while preserving the theoretical validity of the diffusion objective.

Our contributions are threefold: 1) we adapt masked diffusion language modeling for data-restricted settings, exploring multiple noise scheduling strategies including two-mode approaches and different NELBO weighting schemes, 2) we introduce a frequency-informed masking strategy that seamlessly integrates into the diffusion objective while preserving theoretical validity, and 3) we provide comprehensive evaluation on the BabyLM benchmark demonstrating that diffusion-based training achieves competitive performance with established baselines. Our code and weights are made available[1].

2 Related Work

Masked Diffusion Language Modeling Inspired by continuous-time diffusion models (Sohl-Dickstein et al., 2015), diffusion frameworks have emerged as a powerful paradigm for discrete text generation. Austin et al. (2023) introduced `D3PM`, establishing the theoretical foundation for applying diffusion to text, with concurrent work by Hoogeboom et al. (2021) and Campbell et al. (2022) developing discrete and continuous-time formulations. The intersection of diffusion with masked language modeling proved particularly promising. Masked diffusion modeling formulates discrete diffusion as a Markov process with an absorbing state, where tokens replaced by `[MASK]` remain masked in subsequent steps, and the reverse process reconstructs original data from progressively corrupted representations. Sahoo et al. (2024) introduced simplified `MDLMs`, unifying masked language modeling and diffusion through a simplified NELBO expression. This combines bidirectional context benefits with generative training in a unified objective. Similar simplified formulations by Shi et al. (2025) and Ou et al. (2025) demonstrated improved efficiency, with recent work by Sahoo et al. (2025) bridging discrete and Gaussian diffusion for enhanced training techniques.

Masking Strategies for MLMs Several approaches have extended `BERT`'s 15% random token masking (Devlin et al., 2019) with more structured strategies. `SpanBERT` masks contiguous random spans rather than individual tokens and introduces a span boundary objective to predict entire masked spans (Joshi et al., 2020), achieving substantial improvements on span selection tasks. `ELECTRA` replaces tokens with plausible alternatives using a generator-discriminator framework, moving beyond simple masking to token replacement detection (Clark et al., 2020). `RoBERTa` introduces dynamic masking where different tokens are masked across training epochs, in contrast to BERT's static masking approach (Liu et al., 2019). `PMI-Masking` proposes a principled approach based on Pointwise Mutual Information, jointly masking token n-grams with high collocation scores over the corpus (Levine et al., 2020).

Diffusion Models in Data-Sparse Settings Diffusion models face significant challenges when applied to data-constrained image scenarios. Zhu et al. (2022) demonstrated that standard diffusion models suffer from diversity degradation in few-shot settings, leading to overfitting on limited training samples. Wang et al. (2024) identified that image-agnostic Gaussian noise creates uneven adaptation effects and proposed adversarial noise selection for more balanced transfer learning. Lu et al. (2023) showed efficient adaptation through fine-tuning specific attention layers, while Kulikov et al. (2023) explored single-image learning by modeling internal patch distributions. However, these findings focus on vision tasks, leaving diffusion models in data-constrained LM underexplored.

Token Frequency, Weighting and Masking Frequency-based training strategies have emerged to address the imbalance of Zipfian distributions of language tokens. Platanios et al. (2019) demonstrated that curriculum learning based on word frequency can improve sample efficiency in neural machine translation. Bengio et al. (2009) showed that gradually increasing task difficulty, *i.e.*, from

[1] https://github.com/DespoinaKK/babylm-diffusion

frequent to rare tokens, can lead to better convergence and generalization. Importance sampling approaches have been developed to reweight training examples based on token loss (Lin et al., 2024). Recent work has explored adaptive masking strategies that prioritize more salient tokens during training (Choi et al., 2024). However, the application of frequency-based weighting specifically to diffusion models remains unexplored, particularly in data-constrained settings where efficient learning from rare tokens becomes critical.

3 Methodology

3.1 Pretraining

Architecture Our model architecture is a Transformer (Vaswani et al., 2023), based on the `LTG-BERT` model (Samuel et al., 2023), with the attention-gating modifications from (Georges Gabriel Charpentier and Samuel, 2023). To time-condition this model for the diffusion process, we use a timestep embedding and incorporate it with Adaptive Layer Normalization (AdaLN) modulation, following (Peebles and Xie, 2023). This approach enables the model to condition its predictions on the current masking level at diffusion timestep t, allowing it to adapt its behavior across different stages (masking rates) of the diffusion process.

Diffusion Objective Our approach is inspired by both last year's winning `GPT-BERT` method and recent advances in `MDLMs` (Sahoo et al., 2024, Shi et al., 2025). While `GPT-BERT` demonstrates the effectiveness of combining encoding and generative objectives through joint training with MLM and NTP, `MDLMs` results reveal that a single principled diffusion objective can achieve similar dual-purpose training. We adopt the `MDLMs` framework to explore whether this unified approach can be effective in the data-restricted BabyLM setting.

Following (Sahoo et al., 2024), at every training step, a masking rate $1 - \alpha_t$ is sampled from a distribution over $(0, 1)$ for each sequence. Only masked tokens contribute to the cross-entropy loss, and the total objective is a weighted average of MLM losses across different masking levels.

Specifically, in expectation, we optimize the simplified continuous-time NELBO objective from `MDLMs` (Sahoo et al., 2024):

$$\mathcal{L} = \mathbb{E}_q \int_{t=0}^{t=1} \frac{\alpha'_t}{1-\alpha_t} \sum_{\ell=1}^{L} \log \langle \mathbf{x}_\theta^\ell(\mathbf{Z}_t), \mathbf{x}^\ell \rangle \, dt \quad (1)$$

where α'_t denotes the time derivative of the noise schedule α_t, $\mathbf{Z}_t$ represents the masked sequence at time t, $\mathbf{x}^\ell$ the token at position ℓ, $\mathbf{x}_\theta^\ell(\mathbf{z}_t)$ is the model's prediction at that position, and θ the learnable parameters. This formulation provides a principled objective that naturally weights different masking rates according to the diffusion schedule, and involves maximum-likelihood optimization.

Frequency Informed Masking We propose frequency-informed masking that assigns *higher masking probabilities to rare tokens*. This approach prioritizes learning from infrequent but semantically rich tokens rather than common function words. For a given sequence of tokens $\mathbf{Z} = [\mathbf{x}^1, \ldots, \mathbf{x}^L]$ with a pre-assigned masking rate of $1 - \alpha_t$, we follow *a two-step process* to determine the masking probability for each token. First (step-1), we rank tokens based on their global frequency, with rarer tokens receiving higher ranks. These ranks are min-max normalized to produce initial per-token weights $w^\ell \in (0, 1)$, constructing per-sequence weights $\mathbf{w}$. To prevent an over-emphasis on extremely rare tokens, these weights are "softened" by being raised to a power $p < 1$. Our goal is to scale the weights so that they correspond to the tokens' sampling probability. Next (step-2), we apply conditional scaling to these weights to ensure their mean equals the target probability $1 - \alpha_t$.

$$\mathbf{w}_{\text{new}} = \begin{cases} \mathbf{w}^p \, \frac{1-\alpha_t}{\mu} & \text{if } \mu > 1 - \alpha_t \\ -(1 - \mathbf{w}^p)\frac{\alpha_t}{1-\mu} + 1 & \text{otherwise} \end{cases} \quad (2)$$

Each token $\mathbf{x}^\ell$ is then masked with a probability equal to its new weight, w^ℓ_{new}.

This weighting scheme can be naturally extended to a form of curriculum learning (Bengio et al., 2009) by gradually increasing the softening power p from 0 to a value < 1 across training. This process makes the distribution of masking probabilities sharper over time, which forces the model to progressively focus on predicting rarer and more challenging tokens. We note that frequency is only one option for the relative ranking of tokens. In our proposed `MDLMs` framework, any masking strategy can be *flexibly and seamlessly* incorporated.

3.2 Evaluation

We evaluate our framework using the BabyLM Challenge evaluation pipeline, assessing models across linguistic competence, world knowledge,

human-likeness measures, and standard Natural Language Understanding (NLU) tasks. This suite tests both the quality of learned representations and their alignment with human language acquisition.

Zero-Shot Evaluation We evaluate our models on tasks focusing on linguistic performance and understanding, such as BLiMP (Warstadt et al., 2023b) and Blimp Supplement (Warstadt et al., 2023b). Another linguistic test, targeting grammatical generalization is the Derivational Morphology Test (Hofmann et al., 2024), namely the WUG Adjective Nominalization Test, along with a prior contribution, the WUG Past Tense Test (Weissweiler et al., 2023). EWoK (Ivanova et al., 2025) tests the model's *understanding* of the world, including physical concepts and causal relationships. In a similar minimal pair setting, COMPS (Misra et al., 2023) tests inheritance of properties between hierarchical concepts. Entity Tracking (Kim and Schuster, 2023) tests the model's state tracking abilities. In the zero-shot setting, the goal is for the model to assign higher likelihood to the correct sentence, from a group of sentences.

Finetuning Our pretrained model is further finetuned and evaluated on a subset of GLUE (Wang et al., 2019) and SuperGLUE (Wang et al., 2020), testing NLU.

Human-Likeness Alignment with human acquisition is of special interest when training in developmentally plausible settings. We evaluate on a Reading task using data from (de Varda et al., 2024) and on Age of Acquisition (Chang and Bergen, 2022). The derivational morphology tests (Hofmann et al., 2024), (Weissweiler et al., 2023) provide human annotator data, and the higher model with human correlation is favorable.

Evaluation Backend We use the provided MLM backend to estimate pseudo-likelihoods of sentences (Salazar et al., 2020). MDLM can be evaluated with or without time conditioning. Without time conditioning, we set the masking rate to 0, which corresponds to a fully denoised sequence. With time conditioning, we set the masking rate to $1/L$ for a sequence of length L, which matches the expected masking rate when evaluating one token at a time.

3.3 On the MLM evaluation backend

We argue that for MDLMs, the MLM evaluation backend is a rather myopic view of likelihood estimation, as it only focuses on the very last denoising (unmasking) steps, ignoring previous ones. In theoretical contrast to MLMs, MDLMs are generative language models. For MDLMs, perplexity estimation can be viewed as a Monte-Carlo approximation of the diffusion denoising process (Sahoo et al., 2024).

We suggest that a more appropriate evaluation backend would accommodate for the various possible generation trajectories of the same phrase, and thus provide an estimation better aligned with the native diffusion training objective. This approach would require either exhaustive computation, at the expense of exponential compute time, or Monte-Carlo approximation. The latter is practical for perplexity estimation in large texts, but the accompanying non-determinism proves unsuitable for capturing nuances between similar, short sentences. Nonetheless, for the purposes of the BabyLM Challenge, the MLM pseudo-likelihood estimation, utilized for relatively short sentences, offers the advantage of efficient computation, sufficiently good performance, and determinism.

4 Experiments

We briefly describe the training setup and proceed with a series of experiments, ablations, and evaluations which explore different components of the proposed framework and validate the soundness of our method. First, we test different *noise scheduling* options, *e.g.*, uniform and cosine, naturally motivating our submission's adopted approach. We also include an exploration of experimental unimodal and *bimodal gaussian schedules*, ultimately aiming to design a noise schedule that balances the advantages of AR and MLM approaches. Next, we conduct ablation experiments, establishing the benefits of the proposed frequency informed masking method. Finally, we focus on our submission to the BabyLM Challenge, providing implementation details and the full evaluation results.

4.1 Training Setup

Our architecture follows (Charpentier and Samuel, 2024). We use the same tokenization process and optimizer hyperparameters. The training objective aligns with MDLMs' as in Eq. (1). We train our models for 10 epochs on the BabyLM corpus, with a constant sequence length of 128 for ablation studies, and 512 for the submission model.

SCHEDULE	EWoK (↑)	BLiMP (↑)	BLiMP Sup. (↑)
Eval. w/o Time Conditioning			
Uniform	51.98±0.12	77.91±1.35	67.63±3.64
Cosine	52.44±0.24	79.05±0.28	70.74±1.35
Eval. with Time Conditioning			
Uniform	52.16±0.51	77.55±0.55	67.23±0.98
Cosine	52.39±0.48	78.55±0.70	69.41±0.93

Table 1: Performance comparison across different noise schedules, over 5 random seeds. Reported accuracies are field averages. Likelihoods are estimated with the standard MLM Backend.

4.2 Experiments and Ablations

Noise Schedules Table 1 illustrates a comparison between uniform and cosine masking probability schedules. Additionally, we evaluate them with and without time conditioning. We report the zero-shot results for the four configurations.

With the uniform noise schedule all masking rates are treated equally in the loss calculation, which leads to weak results. The cosine schedule focuses on lower masking rates, with an average masking rate of *0.36* compared to the uniform schedule's *0.5*. Our experiments show that the cosine schedule's lower masking rates consistently improve the model's performance in zero-shot likelihood estimation tasks, as they provide more fine-grained focus.

Gaussian schedules In the context of finding a noise schedule that more effectively unifies the benefits of MLM and AR modeling within the masked diffusion framework, we experiment with unimodal and bimodal Gaussian noise schedules. This means that the distribution of $1 - \alpha_t$ is normal (or a Gaussian mixture) when t is sampled uniformly. Table 2 presents results of a qualitative comparison of training with a unimodal and a bimodal noise schedule with similar expected masking rates across training. *Unimodal*, is a unimodal gaussian masking strategy, with masking rates coming from a $\mathcal{N}(0.3, 0.1)$ distribution. *Bimodal*, is a mixture distribution $w_1\mathcal{N}(\mu_1, \sigma_1^2) + (1-w_1)\mathcal{N}(\mu_2(\tau), \sigma_2^2)$ where the right mode progresses to higher values over time. In this experiment, the left mode has weight $w_1 = 0.6$, mean $\mu_1 = 0.12$, and standard deviation $\sigma_1 = 0.02$. The right mode has time-varying mean $\mu_2(\tau) = 0.4 + (0.85 - 0.4)(1 - e^{-\tau})$ and standard deviation $\sigma_2 = 0.08$, with τ representing the training progress.

The importance of scaling α'_t Table 2 shows that using the full derivative term α'_t ($\gamma = 1.0$) in the NELBO optimization leads to poor zero-shot results. However, performance improves significantly when we scale down the derivatives with a small power of γ or remove them completely ($\gamma = 0.0$). The Unimodal schedule shows modest improvement, while the Bimodal schedule shows dramatic gains, nearly matching top baseline scores when derivatives are softened. These results demonstrate that scaling the derivative term is essential when training with Gaussian schedules.

SCHEDULE (γ)	EWoK (↑)	BLiMP (↑)	BLiMP Sup. (↑)
Unimodal(1.0)	50.24	55.70	51.92
Bimodal(1.0)	51.10	68.13	63.0
Unimodal(0.1)	50.65	64.34	59.32
Bimodal(0.1)	52.46	79.49	72.81
Unimodal(0.0)	50.34	65.34	58.76
Bimodal(0.0)	52.95	78.28	73.13

Table 2: Qualitative performance comparison across different noise schedules. Reported accuracies are field averages. Likelihoods are estimated with the standard MLM Backend. (γ) denotes the softening power for the derivative factor. Results are run over 1 random seed.

Frequency Informed Masking Table 3 compares our method's performance across two distinct configurations:

- *No Frequency Weighting*: A baseline where tokens are masked with equal probabilities.
- *Frequency Weighting* (FW): Our frequency-informed method is applied with a softening power of $p = 0.02$, progressively (linearly) reaching this value across epochs.

We inspect the performance of these configurations on EWoK, BLiMP, and BLiMP Supplement, and report on the accuracy of the Adjective Nominalization test. All models were trained on a cosine noise schedule, with sequence length 128.

The frequency informed masking in general preserves or boosts performance across tasks, **improving performance on BLiMP Sup. by an absolute 1% point** consistently. On the **Adjective Nominalization** test, we observed high variance across random seeds, so we conducted a paired comparison, measuring the accuracy difference between models of different configurations trained with the same seeds. The FW configuration, evaluated with time conditioning, enhances performance, **improving it by an average of 7.5 percentage points**.

CONFIG.	EWoK (↑)	BLiMP (↑)	BLiMP Sup. (↑)
Eval. w/o Time Conditioning			
`Cosine`	52.44±0.24	79.05±0.28	70.74±1.35
`Cosine + FW`	52.63±0.36	78.92±0.34	71.77±0.86
Eval. with Time Conditioning			
`Cosine`	52.39±0.48	78.55±0.70	69.41±0.93
`Cosine + FW`	52.21±0.47	78.90±0.37	70.65±1.87

Table 3: Performance comparison across different token frequency weighting configurations, over 5 random seeds. The FW configuration uses weights softened by raising the frequency distribution to power $p = 0.02$ before scaling. Likelihoods are estimated with the standard MLM Backend.

4.3 Submission Model

Implementation *Training Recipe:* A BPE tokenizer (Gage, 1994) was trained with a vocabulary of 16384 tokens. The submission models have size equal to 126.6 M parameters and were trained with a fixed sequence length of 512. The batch size was set to 512, and sequences were not packed. Documents exceeding this length were divided into independent segments. The total training duration was 10 epochs, or 7530 training steps.

Diffusion Model: For our submission to the leaderboard we employed a cosine masking schedule, with $a_t = cos(\frac{\pi}{2}(1-t))$. Timestep embedding dimension was set to 128. For the frequency informed masking, we used $p = 0.02$, starting from 0 at epoch 0 and linearly reaching p at the last epoch.

Evaluation We provide[2] the submission's internal evaluation results, comparing them with the scores of the baseline with the maximum average score under the name `Baseline-gpt-bert-base-mixed (mntp)`. Zero-shot results were computed evaluating with the standard MLM backend, without time conditioning.

Our model is competitive with the baseline models, particularly in the Finetuning evaluation suite, where it performs especially well on the MRPC and RTE tasks (Table 5). On certain zero-shot evaluation tasks, the model slightly underperforms the top-scoring baseline (*e.g.* BLiMP Sup., EWoK), while it achieves better performance in Entity Tracking (Table 4). In terms of human likeness measures, the submission outperforms the top baseline in Reading and on the Adjective Nominalization Test (Table 6).

[2]We will further update our results with the stronger bimodal gaussian schedule in our code release.

TASK	TOP BASELINE	SUBMISSION†
Linguistics		
BLiMP	80.5	76.9
BLiMP Sup.	73.0	72.4
World Understanding		
EWoK	52.4	51.8
COMPS	59.7	56.4
Entity Tracking	39.9	40.8

Table 4: Evaluation results for Linguistics and World Understanding tasks; †: results refer to cosine schedule

Natural Language Understanding (Finetuning)		
TASK	TOP BASELINE	SUBMISSION†
BoolQ	73.4	72.2
MNLI	63.4	63.8
MRPC	85.8	88.7
MultiRC	69.8	69.0
QQP	81.2	79.2
RTE	59.0	64.7
WSC	63.5	65.4

Table 5: Evaluation results for Natural Language Understanding tasks; †: results refer to cosine schedule

Human Alignment		
TASK	TOP BASELINE	SUBMISSION†
Reading	6.3	7.4
WUG Adj. N.	41.2	49.6
WUG Past T.	27.1	15.4
AoA	22.3	-22.0

Table 6: Evaluation results for Human Likeness tasks; †: results refer to cosine schedule

5 Conclusions

`MDLMs` emerge as a compelling pretraining paradigm for data-constrained LM environments, demonstrating competitive performance against state-of-the-art baselines. Our findings reveal that the choice of masking strategy and its induced objective weighting critically determines model effectiveness. Specifically, we demonstrate that cosine noise schedules yield substantial performance gains over uniform schedules, while bimodal approaches unlock even greater potential, but may require special weighting in the NELBO. Furthermore, we establish a principled framework for integrating intra-token masking strategies within the diffusion paradigm, maintaining theoretical coherence while

expanding practical applicability. These results position masked diffusion as a viable path forward for efficient language model pretraining, particularly valuable when computational resources or training data are limited.

Limitations

This work represents a conceptual integration of MDLMs into the LTG-BERT model family, doing minimal architectural modifications. Standard implementations of MDLMs often incorporate additional optimizations that can substantially impact performance; such optimizations are not explored here. Furthermore, accurately and efficiently estimating likelihoods for zero-shot tasks with short sequences using conventional diffusion approaches while maintaining low variance remains an open challenge. We hypothesize that, while the current MLM-based likelihood estimation approach captures relative trends well, it may be suboptimal, further undermining the MDLMs performance.

Acknowledgments

This work has been supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program. We acknowledge EuroHPC JU for awarding the project ID EHPC-AI-2024A04-051 access to the EuroHPC supercomputer LEONARDO hosted by CINECA (Italy).

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2023. Structured denoising diffusion models in discrete state-spaces. *Preprint*, arXiv:2107.03006.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. 2022. A continuous time framework for discrete denoising models. *Preprint*, arXiv:2205.14987.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Hyesong Choi, Hyejin Park, Kwang Moo Yi, Sungmin Cha, and Dongbo Min. 2024. Salience-based adaptive masking: Revisiting token dynamics for enhanced pre-training. In *European Conference on Computer Vision (ECCV)*, pages 343–359. Springer.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *Preprint*, arXiv:2003.10555.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 English sentences, aligned with existing EEG and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, and 1 others. 2017. Mapping the early language environment using all-day recordings and automated analysis. 26(2):248–265.

Valentin Hofmann, Leonie Weissweiler, David Mortensen, Hinrich Schütze, and Janet Pierrehumbert. 2024. Derivational morphology reveals analogical generalization in large language models. *Preprint*, arXiv:2411.07990.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions. *Preprint*, arXiv:2102.05379.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *Preprint*, arXiv:2405.09605.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Preprint*, arXiv:1907.10529.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Vladimir Kulikov, Shahar Yadin, Matan Kleiner, and Tomer Michaeli. 2023. Sinddm: A single image denoising diffusion model. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17920–17930. PMLR.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2020. Pmi-masking: Principled masking of correlated spans. *Preprint*, arXiv:2010.01825.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. Rho-1: Not all tokens are what you need. *arXiv preprint arXiv:2404.07965*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Haoming Lu, Hazarapet Tunanyan, Kai Wang, Shant Navasardyan, Zhangyang Wang, and Humphrey Shi. 2023. Specialist diffusion: Plug-and-play sample-efficient fine-tuning of text-to-image diffusion models to learn any unseen style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14267–14276.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2025. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *Preprint*, arXiv:2406.03736.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *Preprint*, arXiv:2406.07524.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. 2025. The diffusion duality. *Preprint*, arXiv:2506.10892.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: Bert meets british national corpus. *Preprint*, arXiv:2303.09859.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. 2025. Simplified and generalized masked diffusion for discrete data. *Preprint*, arXiv:2406.04329.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. *Preprint*, arXiv:1503.03585.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems. *Preprint*, arXiv:1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Xiyu Wang, Baijiong Lin, Daochang Liu, Ying-Cong Chen, and Chang Xu. 2024. Bridging data gaps in diffusion models with adversarial noise-based transfer learning. In *Proceedings of the 41st International Conference on Machine Learning*, pages 1–11. PMLR.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2023b. Blimp: The benchmark of linguistic minimal pairs for english. *Preprint*, arXiv:1912.00582.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schütze, Kemal Oflazer, and David R. Mortensen. 2023. Counting the bugs in chatgpt's wugs: A multilingual investigation into the morphological capabilities of a large language model. *Preprint*, arXiv:2310.15113.

Jingyuan Zhu, Huimin Ma, Jiansheng Chen, and Jian Yuan. 2022. Few-shot image generation with diffusion models. *arXiv preprint arXiv:2211.03264*.

MoEP: Modular Expert Paths for Sample-Efficient Language Modeling

Joonas Tapaninaho
University of Oulu, Faculty of Information Technology and Electrical Engineering, CMVS
Oulu, Finland

Abstract

Training language models under tight compute budgets with small training datasets remains challenging for dense decoder-only Transformers, where every token activates the full stack of model parameters. We introduce *MoEP* (Modular Expert Paths), a sparse decoder-only architecture that enables more selective token activation, which increases model performance and accelerates learning without increasing the total number of parameters. We show that combining model parallelism with Mixture-of-Experts (MoE) style linear projections and a lightweight top-k router outperforms the GPT-2 baseline and stabilizes evaluation performance more quickly.

1 Introduction

Despite the strong dominance of dense decoder-only Transformers, there is noticeable growing interest in exploring alternative architectures, which challenge the assumption that every token must pass through the same full stack of layers or route.

Recent and previous work has examined **sparse activation** [9, 13], **routing-based decoder-only language modeling** [1, 6, 10, 15], and compositional approaches, where models are constructed from **modular components** [14]. These efforts highlight a broader trend to improve efficiency and flexibility by enabling tokens to follow different computation paths.

Our previous work, *PaPaformer* [14], introduced method of remodeling Transformer layers into smaller **parallel subpaths**, which can be used as independently trainable modules. Despite being effective for modularity, PaPaformer required pre-trained paths to outperform the baseline architecture and did not fully exploit the sparsity opportunities offered by parallel paths.

This paper presents *MoEP* (Modular Expert Paths), which adds model sparsity by unifying two forms of routing within a decoder-only language model: (i) **Top-k** token routing across parallel Transformer blocks, and (ii) **Mixture-of-Experts** feed-forward layers based on lightweight linear projections and comparison **SwiGLU** variant. As a result, each token activates only a limited set of parallel blocks and experts in forward-pass, creating more diverse computational pathways while reducing redundancy. In training a load-balanced **auxiliary loss** was used to encourage stable expert and block utilization without collapse.

We train MoEP with the **BabyLM** strict-small track [1] data and used the official evaluation pipeline. MoEP was able to outperform all BabyLM strict-small baseline models, not only GPT-2, which layer structure it follows. In addition, MoEP exhibits earlier learning gains in comparison to GPT-2, which suggest faster learning capabilities. This was achieved even though MoEP did not employ the *PaPaformer* [14] style of modularity, in which independent modules were pre-trained separately.

[1] https://babylm.github.io

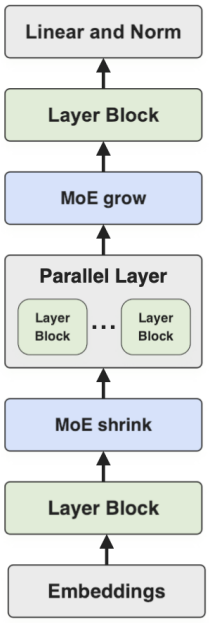


Figure 1: MoEP architecture visualization. N parallel layers are stacked before and after the MoE blocks, whose task is to reduce or increase the hidden dimension to match the layer blocks. In a Parallel layer, the Layer blocks operate on a smaller hidden dimension compared to the individual Layer blocks at the beginning and end of the model.

This work present following contributions, which are summarized below:

(1) Proposes *MoEP*, a modular sparse decoder-only architecture that integrates top-k routing across parallel blocks with MoE style.

(2) Employs the BabyLM evaluation pipeline on the strict-small track to compare MoEP against GPT-2 and other baseline models under matched conditions.

(3) Analyzes fast-eval learning dynamics, showing it earlier stabilization in comparison GPT-2.

(4) Introduces a SwiGLU-based MoEP variant, whose learning behavior is more similar to GPT-2, but which struggles to match its performance.

Proceedings of the First BabyLM Workshop, pages 552–559
November 8, 2025

2 Related Works

2.1 Sparse and Routing-Based Models

Mixture-of-Experts (MoE) architectures [9, 13] introduced sparse token-wise routing within feed-forward layers, enabling models to increase capacity without a proportional increase in computational density. Follow-up works such as GLaM [6], DeepSeek-V2 [15], and OLMoE [10] extended this idea with improved routing strategies. More recently, approaches like MoR [1] explored layer-level routing, where different tokens may skip or use fewer layers. These works reflect a broader trend toward architectures that diversify token computation paths beyond uniform dense stacks.

Our work aligns with this trajectory but integrates routing both across parallel Transformer blocks and within MoE experts.

2.2 Parallel Architectures

As alternative to dense Transformer architecture design, some works have explored parallelization as purpose to increase expressiveness or efficiency. PaLM [4] introduced pathway-based scaling, while Branchformer [11] combined MLP and attention to parallel components. Our prior work, PaPaformer [14], proposed a alternative approach, combining independently trained parallel paths into larger composite models. MoEP is build on this line by maintaining parallelism but coupling it with more MoE style top k routing.

2.3 Tiny Language Models

Evaluating new architectures at small scale has become increasingly important, as recent results show that novel architectural methods can notably improve model performance, while the size of current Large Language Models (LLMs) limits the threshold for exploring such innovations. However, small datasets such as **TinyStories** [8] and **BabyLM** [3] enable rapid iteration with models under 100M parameters. The BabyLM challenge explicitly emphasizes architectural innovations under a 100M and 1B-word budget and provides a comprehensive evaluation-pipeline[2].

MoEP is designed within this paradigm: small enough for fast training, yet still large enough to be reasonably evaluated on benchmark suites such as BLiMP [19] and SuperGLUE [18].

2.4 Decoder-Only Baselines

Dense decoder-only Transformers have long been the standard for autoregressive modeling, exemplified by **GPT-2** [12], **GPT-3** [2], **LLaMA-2** [17], and **LLaMA-3** [7], as well as Google's **Gemini** models [16]. These baseline models provide strong performance, but process every token through the same layers and routes. MoEP is directly compared against GPT-2 under matched data, optimization settings, and training conditions to isolate the effect of modular sparse routing.

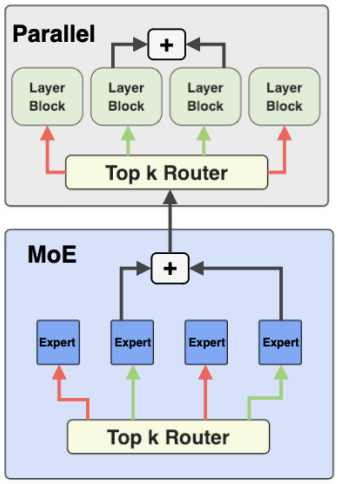


Figure 2: Overview of MoEP routing structure. Each token is routed through a sparse subset of experts in a Mixture-of-Experts (MoE) block, followed by a top-k routed selection of Layer Blocks in a Parallel stack. The routers select k components, whose outputs are summed. This design allows different tokens to follow distinct computation paths through both experts and parallel layers.

3 Methodology

3.1 Overview of MoEP architecture

MoEP and **MoEP-SwiGLU** (see Figure 1), is a decoder-only model that interleaves two standard (dense) *Large Layers* with a sparse middle stack: **Layer Block** (full size) → **MoE** (shrink) → **Parallel Layer** N times with top-k routing → **MoE** (grow) → **Layer Block**.

The first full size **Layer Block** operates at a higher hidden dimension d_{L}. A shrinking **MoE Block** uses E experts with top-k routing, where experts are either simple Linear layer or SwiGLU, which map projection in to smaller hidden dimension d_{P} suited for the routed parallel stack (see Figure 2). **Parallel Layer** uses top-k routing among P **Layer Blocks**, which uses hidden dimension d_{P}. After N Parallel Layers, a growing **MoE Block** projection maps back from d_{P} to d_{L} before the second Large Layer.

3.2 Parallel Layers (Block Routing at Smaller Dimension)

Each **Parallel Layer** contains P Transformer blocks $\{B_1, B_2, \ldots, B_K\}$, which are architecturally equivalent to the full size **Layer Block**, but operates at the reduced dimension d_{P} (same sublayer structure, distinct parameters) and **Router**, which is simple Linear Layer size d_{P} x P. In token-level, **Router** applies **top-k** selection among P **Layer Block** and routed inputs are summed together. This routing method allows

[2]https://github.com/babylm/evaluation-pipeline-2025/

different tokens traverse with different subsets of blocks within each Parallel Layer.

Stacking N **Parallel Layers** yields a deep routed path in compact dimensions.

3.3 MoE Projections (Shrink and Grow)

The two **MoE Block** projections implement the dimensionality transitions:

$$\text{shrink: } d_{\text{L}} \to d_{\text{P}}, \qquad \text{grow: } d_{\text{P}} \to d_{\text{L}}.$$

Each **MoE Block** consists of E experts and a token-level top-k routing over experts. In the base **MoEP** model, experts are simple *linear* projections and in **MoEP-SwiGLU**, experts use *SwiGLU*-based feed-forward projections.

3.4 Routing Objective and Training Loss

To avoid expert and block collapse, in training phase we used a standard load-balancing regularizer.

Let p_i denote the average routing probability assigned to block or expert i over a batch. The balancing term is

$$\mathcal{L}_{\text{balance}} = -\sum_i p_i \log p_i,$$

computed separately for block routing and expert routing. The total objective is

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda^{\text{block}} \mathcal{L}^{\text{block}}_{\text{balance}} + \lambda^{\text{expert}} \mathcal{L}^{\text{expert}}_{\text{balance}},$$

where $\mathcal{L}_{\text{CE}}$ is the next-token cross-entropy loss and λ learning weight.

4 Training and Experimental Setup

Training Data

For training, we used only the **BabyLM** [3] strict-small dataset, without any additional text preprocessing. The corpus contains a little over 10 million words, drawn from curated English sources including **CHILDES**, **BNC Spoken**, **Gutenberg**, **OpenSubtitles**, **Simple Wikipedia**, and **Switchboard**. No external data were added, in order to ensure direct comparability with the baseline submissions in the track.

Tokenization

All models were trained with the same **GPT-2** style **byte-pair encoding** (BPE) tokenizer. We use a fixed vocabulary of 16K tokens, trained on the **BabyLM** strict-small corpus. This size balances compactness with adequate coverage of rare subwords. The tokenizer follows a similar pattern-recognition strategy as **babylm-baseline-10m-gpt2** [3], avoiding the need for training data preprocessing and ensuring maximal similarity with the **BabyLM** baseline models.

Training Procedure

MoEP, **MoEP-SwiGLU**, and **GPT-2** baseline model were trained from scratch under identical training settings and with causal language modeling objective. We used **AdamW** with **cosine learning rate decay** for stable model training with standard **dropout** and **weight decay** regularization. Initially, we pre-tokenized the training data with a **stride of 128**. During training, examples were randomly sampled from the full pre-tokenized dataset using an epoch-based shared seed, ensuring that all models were trained on the same examples.

Each model was trained for 10 epochs, with training in each epoch stopped after the model had seen approximately 10M words.

Checkpoints were saved every 1M words up to 9M words, and subsequently every 10M words up to 100M words. After training, we ran fast evaluation on all checkpoints, and the final model weights were taken from the checkpoint with the best evaluation performance. These weights were then used for full evaluation. **MoEP** and **GPT-2** achieved their best accuracy at 30M words, while **MoEP-SwiGLU** reached its peak after 80M words.

A model hyperparameters (hidden dimension, number of layers, parameter counts) were selected to match with BabyLM baseline models and these are listed in Appendix A and Appendix B.

Evaluation Protocol

Evaluation followed the official BabyLM pipeline [3]. Zero-shot evaluation included **BLiMP**, **EWOK**, **WUG**, and other tasks, with the full list available in the evaluation pipeline documentation[4]. For tasks involving finetuning (e.g., **MNLI**, **QQP**, **RTE**), the **BabyLM** evaluation pipeline supplied both training data and default finetuning parameters, which we adopted directly.

Environment

All experiments were conducted with single NVIDIA A100 GPU in CSC's Puhti supercomputing environment [5]. Training a single model for 10 epochs required approximately 1–2 hours, a duration that could be further reduced with code optimizations. All code is implemented using **PyTorch** and **Hugging Face** libraries and released for reproducibility [5] and model is directly downloadable in Hugging Face [6].

5 Results

5.1 Evaluation Scores

As table 1 shows, MoEP achieved the highest performance across all models, including the official **BabyLM** baselines under the strict-small track, when the **AoA** task score was included in the **Macro Average**. Even when excluding AoA from the macro average, MoEP still outperformed the our and the official BabyLM GPT-2 baseline, which we consider our primary comparison point due to the similarity (MoEP-SwiGLU) or full correspondence (MoEP) in layer architecture. MoEP also obtained the best score in five individual tasks, the highest count among all models evaluated.

Among our models, the GPT-2 variant slightly outperformed the **BabyLM GPT-2 baseline** in macro average without **AoA**, reaching performance comparable to **MoEP**. However, subsequent analysis revealed a key distinction - **MoEP** extracted useful patterns earlier during training. This indicates that modular sparse routing can provide better sample efficiency, even if final scores converge to similar levels.

[3] https://huggingface.co/BabyLM-community/babylm-baseline-10m-gpt2

[4] https://github.com/babylm/evaluation-pipeline-2025/

[5] https://github.com/Jtapsa/BabyLM-2025

[6] https://huggingface.co/Jtapsa/moep

Model	Zero-shot Tasks							Finetuned Tasks							Macro
	BLiMP	EWOK	Entity	WUG	Comps	Reading	AoA	BoolQ	MNLI	MRPC	MultiRC	QQP	RTE	WSC	Avg
Our Models															
GPT-2	59.70	**57.85**	13.15	36.00	51.20	6.40	–	67.50	49.10	69.60	66.70	71.55	**62.60**	63.45	48.10 –
MoEP [7]	59.15	50.20	**35.65**	33.00	50.70	**6.70**	53.70	66.20	48.10	70.10	64.50	70.75	**62.60**	**67.30**	49.00 **44.50**
MoEP [8] (SwiGLU)	60.35	49.50	17.10	36.50	51.35	6.60	–	66.30	48.30	70.60	67.25	69.40	54.70	61.55	47.70 –
HF Baselines															
GTP-2 [9]	61.75	49.90	13.90	30.55	51.70	6.50	11.7	52.10	33.10	67.60	57.50	63.60	56.10	61.50	46.60 37.40
GPT-BERT [10] (causal)	**67.45**	49.50	34.60	36.05	52.80	**6.70**	-3.90	**68.10**	46.90	74.50	**68.30**	76.70	56.10	65.40	**54.10** 41.20
GPT-BERT [11] (focus-causal)	62.35	49.5	31.10	32.70	**52.90**	6.50	3.8	67.60	51.80	**78.90**	67.40	**77.40**	57.60	61.50	53.65 40.00
GPT-BERT [12] (mixed-causal)	65.60	50.20	25.40	**48.50**	25.00	6.40	14.50	66.70	**53.30**	77.50	67.00	76.60	55.40	63.50	52.40 39.20

Table 1: Evaluation scores on BabyLM tasks for our models (top) and Hugging Face baseline models (bottom). Two macro averages are reported: the first excludes the AoA result obtained from the Hugging Face leaderboard, while the second represents the overall text-average. In table, BLiMP refers to the average over BLiMP and BLiMP-supplement, WUG corresponds to the average of Wug Adjacency and Wug Past Tense, and Readings is the average of Eye Tracking and Self-Paced Reading tasks.

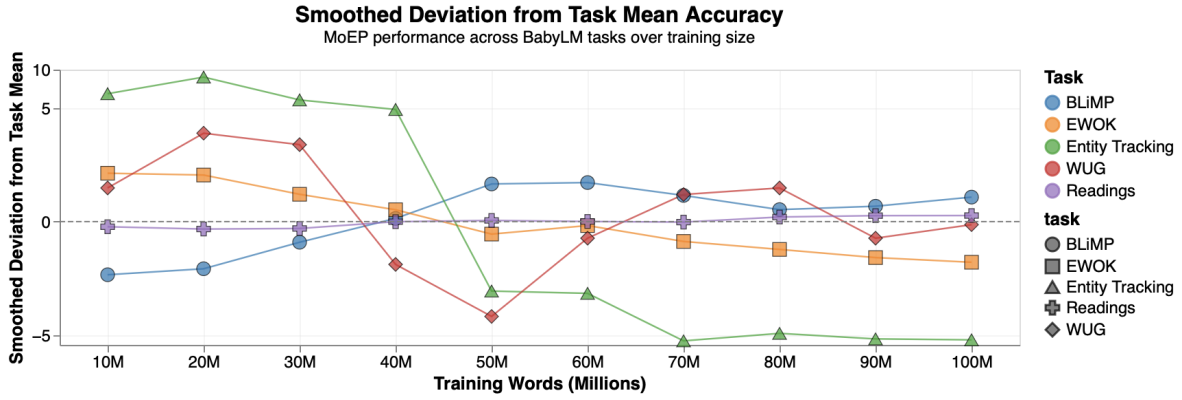


Figure 3: Smoothed deviation from task mean accuracy for MoEP. The dashed origin line represents the average result, while smoothed deviation shows the task accuracy at specific checkpoint relative to the mean.

By contrast, **MoEP-SwiGLU** did not reach the same level of performance. This suggests that lightweight linear experts are more effective at the small scale, whereas **SwiGLU** based feed-forward experts require longer training to stabilize and still achieve lower overall scores compared to the other models.

Note that our **GPT-2** and **MoEP-SwiGLU** results do not include **AoA** scores, which are provided in the official **BabyLM** leaderboard.

5.2 Analysis of Training Development

To better understand how each model architecture learns over training, we analyze their fast-evaluation scores across checkpoints. In the following training dynamics analysis, **BLiMP** refers to the average over BLiMP and BLiMP-supplement, **WUG** corresponds to the average of Wug Adjacency and Wug Past Tense, and **Readings** is the average of Eye Tracking and Self-Paced Reading tasks.

MoEP

Figure 3 presents results for the **MoEP** model. Compared to **GPT-2** (see Figure 4), **MoEP** exhibits more comprehensive early learning, reaching peak performance at the 30M checkpoint, where nearly all task scores are at or above their task-specific means. After 90M words, deviations regress toward zero, with **Entity Tracking** in particular stabilizing well below the mean. This indicates that **MoEP** quickly learns to achieve near-optimal evaluation performance but later begins to overfit, leading to diminished generalization. The

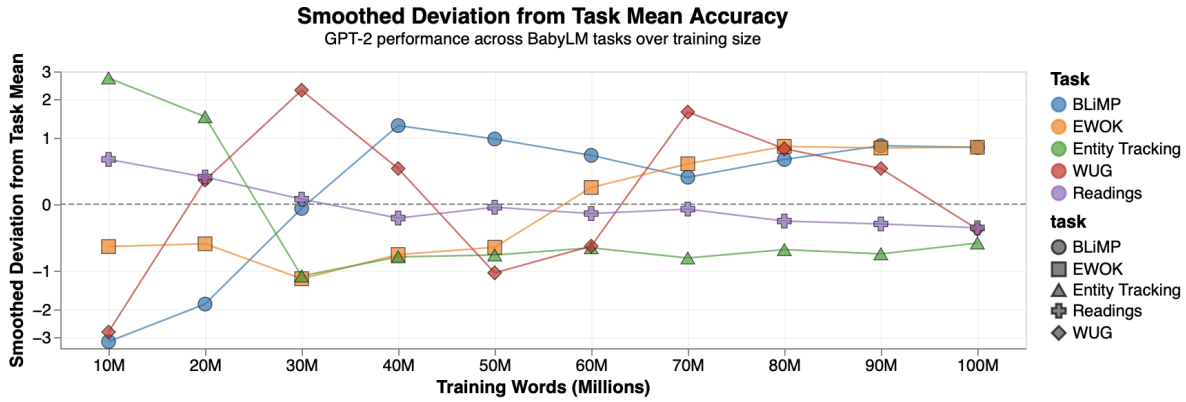


Figure 4: Smoothed deviation from task mean accuracy for GPT-2. Where The dashed origin line represents the average result, while smoothed deviation shows the task accuracy at specific checkpoint relative to the mean.

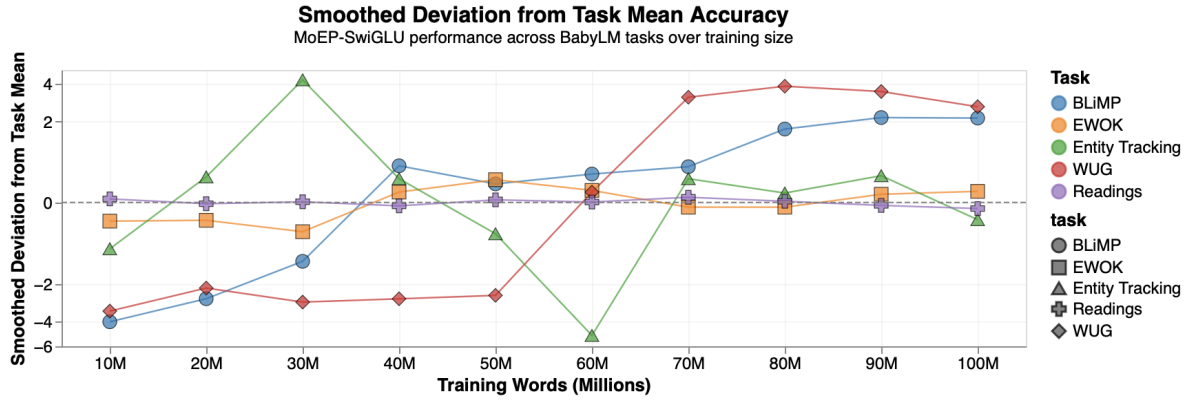


Figure 5: Smoothed deviation from task mean accuracy for MoEP-SwiGLU. The dashed origin line represents the average result, while smoothed deviation shows the task accuracy at specific checkpoint relative to the mean.

pattern highlights that modular routing accelerates initial pattern discovery but may not sustain improvements throughout training.

GPT-2

Figure 4 shows the **GPT-2** baseline smoothed task-mean fast-evaluation results. Unlike **MoEP**, once **GPT-2** reaches its best performance at the 30M checkpoint, it does not stabilize as quickly but continues to improve on certain tasks. On the other hand, after the 70M checkpoint, **WUG** begins to decline and shows no clear signs of stabilization. This reflects a key tradeoff of dense architectures: the model reaches its best scores on different evaluation tasks at different checkpoints rather than converging to a consistent stable state.

MoEP-SwiGLU

Unlike **MoEP**, **MoEP-SwiGLU** (see Figure 5) shows a development more similar to **GPT-2**. The model exhibits strong late-phase improvements on **WUG** and **BLiMP**, with performance rising steadily after 60M words, while other tasks begin to stabilize. **MoEP-SwiGLU** reaches its best performance at the 80M checkpoint, much later than the other models. As with **MoEP** and **GPT-2**, **Entity Tracking** shows strong instability, where early gains at the first checkpoints collapse sharply afterward. These results suggest that SwiGLU-based experts can improve performance on certain tasks, while other evaluations stabilize without the declines observed elsewhere.

Comparative Trends

MoEP learns rapidly during its early checkpoints, showing early specialization particularly on **Entity Tracking** and **WUG**. After the peak at the 30M checkpoint, however, subsequent checkpoints achieve notably lower evaluation scores.

MoEP-SwiGLU achieves the strongest late-phase gains on **WUG**, but this comes at the cost of weaker performance on **Entity Tracking** and **BLiMP**.

GPT-2 shows steadier learning and after reaching its peak performance it experiences fewer dramatic changes in later checkpoints compared to the MoEP variants.

In conclusion, these metrics illustrate that sparse modular routing can accelerate early learning but also introduces instability. The choice of expert type (**linear** vs. **SwiGLU**) further shifts the balance between stability and specialization.

6 Discussion

Limitations

Despite promising results, **MoEP** and **MoEP-SwiGLU** were trained only on a small dataset. It therefore remains unclear whether scaling up the model size and training data would preserve their relative performance compared to **GPT-2**. Within **BabyLM**, where the training corpus and patterns to be learned are relatively simple, smaller-dimensional parallel blocks can capture these patterns as effectively as dense **GPT-2** layers. With more complex data, however, parallel layers may no longer operate effectively at reduced dimensionality, forcing an increase in total parameters that could exceed those required by a dense **GPT-2** to learn the same patterns.

This work also did not include a detailed analysis of expert and block routing. Routing dynamics may have influenced the fast-evaluation results, in way that faster learning observed at early checkpoints could be a consequence of more flexible routing, while the current load-balancing regularizer may have forced overly uniform usage, negatively impacting final evaluation scores. Finally, due to the small model and dataset scale, the present study focused on evaluation benchmarks only and did not investigate generation capabilities. Such analysis might reveal additional differences between sparse **MoEP** and dense **GPT-2** architectures.

Architectural Takeaways

The experiments suggest three main lessons:

- Sparse block and expert routing accelerates early learning, but overall evaluation scores decline afterwards and do not fully recover. This drop is driven by permanent degradation on certain tasks, which lowers the aggregate performance.
- SwiGLU experts increase task specialization and yield late-phase gains, but also amplify volatility and fail to achieve overall results comparable to linear experts.
- Parallel models can match or even outperform dense architectures in the BabyLM strict-small setting. This shows that lower-dimensional sparse paths are sufficient to capture relatively simple language patterns.

Future Work

Future extensions of MoEP could explore:

- Scaling the number of parallel blocks and MoE experts beyond the current four to further increase model sparsity.
- Testing alternative expert architectures in the MoE projections.
- Exploring different load-balancing regularization strategies and analyzing their effects on learning dynamics and evaluation performance.
- Extending evaluation to larger and more complex training datasets to test whether MoEP retains its ability for fast learning and stable evaluation performance.

7 Conclusion

We presented *MoEP*, a sparse decoder-only architecture that combines top-k routing across parallel blocks with linear and feed-forward Mixture-of-Experts projections, allowing the model to flexibly adjust dimensionality across layers.

Within the BabyLM strict-small track, **MoEP** outperformed all official BabyLM baseline models, not only **GPT-2** (the architecture on which it is based), even though **GPT-2** itself was the weakest among the BabyLM baselines.

Our analysis demonstrates a tradeoff in which sparse modular routing accelerates early learning but also introduces higher training variance, with performance often peaking early and then declining. The **MoEP-SwiGLU** variant further showed that expert design directly influences both learning speed and stability. This may be due to the increased parameter size, which is an effect of the **MoEP-SwiGLU** expert design, although the task based learning behavior is neither similar nor stable compared to **MoEP**.

These findings suggest that layer-level sparse, routing-based architectures provide a viable path toward sample-efficient language modeling, even under small-scale budgets. Future work will focus on improving learning stability, optimizing sparse computation, and extending modular expert routing to larger-scale settings.

Acknowledgments

This work was made possible through computation environments provided by the University of Oulu ICT services. I would like to thank Prof. Mourad Oussalah and MSc. Moinul Islam for their valuable feedback and helpful suggestions, which contributed to the development of this research.

References

[1] Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. 2025. Mixture-of-Recursions: Learning Dynamic Recursive Depths for Adaptive Token-Level Computation. arXiv:2507.10524 [cs.CL] https://arxiv.org/abs/2507.10524

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[3] Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM Turns 3: Call for papers for the 2025 BabyLM workshop. arXiv:2502.10645 [cs.CL] https://arxiv.org/abs/2502.10645

[4] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, and et al. 2022. PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311 [cs.CL] https://arxiv.org/abs/2204.02311

[5] CSC – IT Center for Science. [n. d.]. Puhti Supercomputer. https://docs.csc.fi/computing/systems-puhti/. Accessed: 2025-05-06.

[6] Nan Du, Le Hou, Aitor Zhang, Anton Bakhtin, Nathan Scales, Zhifeng Dai, Xin Li, Shixiang Xie, William Fedus, Mostafa Dehghani, and et al. 2022. GLaM: Efficient Scaling of Language Models with Mixture-of-Experts. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=k7K6kB9td9

[7] Abhimanyu Dubey and et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).

[8] Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How Small Can Language Models Be and Still Speak Coherent English? arXiv:2305.07759 [cs.CL] https://arxiv.org/abs/2305.07759

[9] William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 8473–8483. https://proceedings.neurips.cc/paper/2021/hash/2c5dc10619a37b0c79ef595e0bda0592-Abstract.html

[10] Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe

Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. 2025. OLMoE: Open Mixture-of-Experts Language Models. arXiv:2409.02060 [cs.CL] https://arxiv.org/abs/2409.02060

[11] Yifan Peng, Siddharth Dalmia, Ian Lane, and Shinji Watanabe. 2022. Branchformer: Parallel MLP-Attention Architectures to Capture Local and Global Context for Speech Recognition and Understanding. arXiv:2207.02971 [cs.CL] https://arxiv.org/abs/2207.02971

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* (2019).

[13] Noam Shazeer, Azalia Mirhoseini, Andrew Maziarz, Krzysztof Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=B1ckMDqlg

[14] Joonas Tapaninaho and Mourad Oussala. 2025. PaPaformer: Language Model from Pre-trained Parallel Paths. arXiv:2508.00544 [cs.CL] https://arxiv.org/abs/2508.00544

[15] DeepSeek Team. 2024. DeepSeek V2: Scaling Vision-Language Models with Mixture of Experts. arXiv:2401.00733 [cs.CL]

[16] Gemini Team and et al. 2023. Gemini: A Family of Highly Capable Multimodal Models. *Google DeepMind Technical Report* (2023).

[17] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Sharan Batra, Akshat Bhargava, Shruti Bhosale, et al. 2023. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[18] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, Vol. 32.

[19] Alex Warstadt, Yining Cao, Jun Ho, Ellie Pavlick, and Samuel R Bowman. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics* 8 (2020), 377–392.

A Model Hyperparametrs

Comparison of architectural hyperparameters across model variants.

Hyperparameter	GPT-2	MoEP	MoEP SwiGLU
Vocabulary size	~ 16K	~ 16K	~ 16K
d_{model}	384	384 / 192	384 / 192
Layers	12	2 / 10	2 / 10
Parallel blocks	-	4	4
Heads	6	6 / 3	6 / 3
Head dimension	64	64	64
FF multiplier	4	4	4
FF type	MLP	MLP	SwiGLU
MoE FF type	-	Liner	SwiGLU
N experts	-	4	4
Top k	-	2	2
Normalization	LN	LN	LN
Attention	MHA	MHA	MHA
Train seq len	512	512	512
Total Parameter (millions)	28M	28M	38M

Table 2: Architectural hyperparameters of GPT-2, MoEP, and MoEP-SwiGLU.

B Training Setup

Detailed training configurations for all models.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	3×10^{-4}
Batch size	16
Training epochs	10
Gradient accumulation steps	1
Weight decay	0.1
Adam betas	(0.9, 0.95)
Adam epsilon	1×10^{-8}
Scheduler type	Cosine
Warmup steps	800
Random seed	42

Table 3: Training setup and optimization parameters.

RecombiText: Compositional Data Augmentation for Enhancing LLM Pre-Training Datasets in Low-Resource Scenarios

Alexander Tampier[*], **Lukas Thoma**[*○•], **Loris Schoenegger**[*•], **Benjamin Roth**[*△]
[*]Faculty of Computer Science, University of Vienna, Vienna, Austria
[○]Department of Linguistics, University of Vienna, Vienna, Austria
[•]UniVie Doctoral School Computer Science, Vienna, Austria
[△]Faculty of Philological and Cultural Studies, University of Vienna, Vienna, Austria
Correspondence: alexander.tampier@univie.ac.at

Abstract

We introduce RecombiText Augmentation (RTA), a novel purely statistical NLP method for compositional data augmentation for data-efficient LLM pre-training in low-resource scenarios. RTA identifies lexically and semantically similar sentences within the corpus and generates synthetic sentence pairs from them while preserving underlying patterns from the corpus. We pre-train GPT-2 and RoBERTa language models on a domain-specific, low-resource corpus of 10 million words, with different proportions of augmented data. We compare our RTA-augmented model variants to a baseline model trained on the full original dataset. Zero-shot results show that the language models pre-trained on synthetic data improve in entity tracking, self-paced reading, and morphological generalization benchmarks. In other tasks, the performance is comparable to the baseline model. We demonstrate that it is possible to expand low-resource datasets by two- to four-fold without compromising benchmark performance, solely through statistical processing of the available data.

1 Introduction

Large language models (LLMs), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), GPT-3 (Brown et al., 2020), and Chinchilla (Hoffmann et al., 2022), are large-scale language models based on the Transformer architecture from Vaswani et al. (2017) that have achieved remarkable performance across various Natural Language Processing (NLP) tasks. However, success is based on extensive training data, often hundreds of billions of words. For example, GPT-3 (Brown et al., 2020) was trained with 570 GB of text after filtering. This typically results in high computational costs, as well as a dependency on vast amounts of available training data in the respective language or domain. However, large amounts of data are not always available in all languages or domains, which limits the applicability of current language model pre-training for low-resource scenarios (Charpentier et al., 2025; Hedderich et al., 2020). In contrast, human language acquisition is far more efficient. For example, children can fully learn a language by the time they reach puberty, even though they are only exposed to 3 to 11 million words per year (Warstadt and Bowman, 2022; Warstadt et al., 2025).

This discrepancy underscores the limitations of current LLM pre-training and emphasizes the need for data-efficient pre-training in low-resource scenarios. Data augmentation (DA) offers a promising solution for efficiently utilizing available training data by generating synthetic examples from existing datasets (Warstadt et al., 2025; Hu et al., 2024). While DA is well-explored in computer vision and downstream NLP tasks (Feng et al., 2021), its application for LLM pre-training in low-resource scenarios is less explored (Warstadt et al., 2025; Hu et al., 2024). Recent efforts show that DA methods can improve model performance. However, many rely on generative models or auxiliary text data beyond the limited domain training set (Theodoropoulos et al., 2024; Haga et al., 2024; Edman et al., 2024; Zhang et al., 2023; Lyman and Hepner, 2024), thereby limiting their suitability for scenarios in which such auxiliary data is not available.

To address this gap, we propose RecombiText Augmentation (RTA). This novel statistical compositional DA method leverages information retrieval techniques and combines lexical and semantic similarity by utilizing a one-point crossover, a concept inspired by genetic algorithms (Goldberg, 1989). Since RTA relies exclusively on the corpus itself, it is independent of models trained on additional text and is therefore ideal for truly low-resource scenarios. RTA generates synthetic sentences in four corpus-dependent phases:

i. Generating corpus-based embeddings
ii. Selecting matching candidates

Proceedings of the First BabyLM Workshop, pages 560–577
November 8, 2025

iii. Identifying pivot elements with sliding context windows
iv. Applying a one-point crossover to create synthetic sentence pairs.

Experiments on a 10-million-word corpus, acting as a domain-specific low-resource scenario, show that the language models that were trained with RTA-augmented datasets improve the most for performances in zero-shot Entity Tracking (Kim and Schuster, 2023; Charpentier et al., 2025), Self-paced Reading (de Varda et al., 2024), and morphological generalization tasks (WUGs) compared to the baseline.

We demonstrate that purely statistical compositional data augmentation can effectively enhance the language model pre-training dataset in low-resource scenarios without incurring any significant losses in evaluation.

2 Related Work

Good-Enough Compositional Data Augmentation (GECA) (Andreas, 2020) is another compositional data augmentation algorithm for language modeling. GECA identifies and swaps substitutable fragments from sentences that share similar local environments to generate synthetic compositional examples, thereby enabling compositional text recombination without relying on models trained on additional text. In contrast to GECA, our RTA method focuses on lexical and semantical similarities.

Related data augmentation methods for efficient language modeling in low-resource scenarios include using word embeddings from Mikolov et al. (2013) for word substitutions within the training data and external treebanks to ensure grammatical correctness, as proposed by Lyman and Hepner (2024). Haga et al. (2024) generates artificial variation sets that mimic children's speech to produce paraphrased utterances using a pre-trained language model trained on extensive external data. Theodoropoulos et al. (2024) trains a decoder on subsets of the TinyStories dataset (Eldan and Li, 2023) to generate synthetic examples. While these methods enhance performance in low-resource scenarios, many rely on models that were trained on additional text (Lyman and Hepner, 2024; Haga et al., 2024; Edman et al., 2024; Zhang et al., 2023; Theodoropoulos et al., 2024).

3 RecombiText Augmentation

RTA relies exclusively on information from the training corpus, addressing the limitations of methods that depend on resources trained with additional text. Our method generates synthetic sentence pairs based on lexically and semantically similar sentences from the corpus.

3.1 Intuition

RTA is based on the idea of ad-hoc information retrieval (IR), where a user sends a query in natural language and receives relevant results from a collection of documents (Jurafsky and Martin, 2019). Furthermore, it is assumed that sentences that share similar local environments can be swapped to some extent. Based on this, we select a reference sentence that represents the query and perform a hybrid search for lexically and semantically similar sentences within the corpus. We then re-rank them, similar to hybrid search techniques. We borrow the idea from genetic algorithms (Goldberg, 1989) and use a one-point crossover to cut and swap the sentence fragments. To determine the intersection of the reference and candidate sentences, we must assess the pivot elements in each respective sentence. For this purpose, a sliding context window is employed based on semantic similarity and term importance within both sentences. Figure 1 shows the intuition behind the RTA algorithm to create new sentence pairs.

3.2 Algorithmic Formulation

The RTA algorithm operates in four phases using corpus statistics and accessing only the available training data. It combines information retrieval techniques for candidate matching, statistical embeddings for semantic similarity, and a genetic algorithm-inspired crossover for augmentation. The process is divided into four main phases: (i) Word and Sentence Embeddings, (ii) Candidate Selection, (iii) Context Window, and (iv) Crossover Operation. The source code is publicly available [1].

Word and Sentence Embeddings Word embeddings with Global Vectors for Word Representation (GloVe) (Pennington et al., 2014), and sentence embeddings with unsupervised smoothed inverse frequency (uSIF) (Ethayarajh, 2018) are created beforehand. Both methods use the available corpus training data. The sentence embeddings are

[1] https://github.com/luciendgolden/RTA

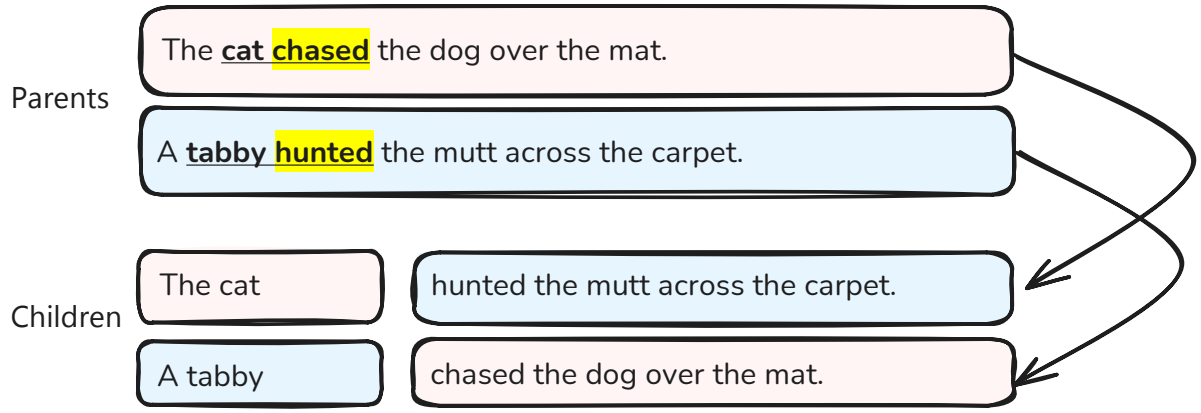


Figure 1: The idea is to create new synthetic sentences based on lexical and semantic similarity from parts of sentences using a one-point crossover. This involves searching for a semantic context window that maximizes the IDF-weighted cosine similarity between a reference and candidate sentence, to determine the pivot elements at which the sentences can be cut and swapped. In this example, the semantic context window size is $W = 2$

stored in the IR system for efficient use of semantic search.

Candidate Selection To generate the synthetic data, a query sentence q is randomly selected from the specified corpus. For a query sentence q, lexically and semantically similar candidates are retrieved. Lexical matches are determined via Best Matching 25 (BM25) (Robertson and Zaragoza, 2009) and give ranking $\mathcal{R}_{BM25}$, which is an extended TF-IDF variant where TF-IDF is the product of two terms, namely the term frequency (TF) and the inverse document frequency (IDF). Semantic matches are determined via k-Nearest Neighbors (kNN) with cosine similarity on sentence embeddings and give ranking $\mathcal{R}_{kNN}$, which is used for the semantic classification of the candidates. These rankings are fused using Reciprocal Rank Fusion (RRF) (Cormack et al., 2009) with constant k_{RRF} to produce $\mathcal{R}_{RRF}$, forming the result set $\mathcal{R}_{RRF} = \{d \mid d \in \mathcal{R}_{BM25}\} \cup \{d \mid d \in \mathcal{R}_{kNN}\}$ where d represents the document sentence. A candidate m is then selected from the top k_{top} via top-k sampling with softmax probabilities modulated by temperature T. From the amount of synthetic text generated $\mathcal{D}_{aug}$ relative to the proportional baseline dataset $\mathcal{D}_{base}$ we get an augmentation ratio r, which defines the maximum frequency of use for q and m. Therefore, in the dataset variant that combines 1 million words from the baseline dataset with 9 million words from synthetic text, $\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$, each q and m sentence can be selected up to 9 times, in the $\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$ variant, up to 3 times, and in all other variants only once.

Context Window To ensure linguistic consistency of the generated sentences after the crossover operation, pivot elements are identified where the query sentence and the selected candidate exhibit high semantic similarity in terms of context. This is achieved using a sliding context window for q, m where sequences of the two sentences are compared based on the importance and semantic equivalence. To avoid placing too much importance on words that are often insignificant, the respective words within the context window are weighted by their IDF values. The context window is defined by a fixed size W where W determines the number of words within the window. Within the respective windows W_q, W_m, the words with the highest similarity and importance are searched for.

For each possible starting position in both sentences, a window of size W is defined, which starts at positions i in Q and j in M where Q and M are tokenized versions of the query sentence q and candidate sentence m. Therefore, we define a context similarity $S_{\text{ctx}}(i, j)$ between the corresponding windows W_q, W_m. Equation 1 shows the context similarity calculation.

$$S_{\text{ctx}}(i, j) = \frac{\sum(\cos_k \times \text{idf}_k)}{\sum \text{idf}_k} \tag{1}$$

where k is each token pair within the windows W_q and W_m and $\cos_k$ is the respective cosine similarity between the word embeddings in W_q and W_m. A context window is only accepted if it is above the threshold τ_{window}. Within the best window, find the pair with the maximum similarity $\cos_k \times \text{idf}_k$, which acts as pivot elements for the

following crossover operation.

Crossover Operation One-point crossover is applied at pivot elements to produce new synthetic sentence pairs $\tilde{q}, \tilde{m}$ where the splitting takes place at pivot elements. The algorithm also defines and considers various edge cases, such as when identical sentences are produced. If such edge cases occur, additional retries are executed. If the algorithm fails several times, a new attempt with a randomly new query sentence q is executed.

4 Experimental Setup

The experiments are carried out in four stages. Firstly, the generation of the different training data variants. Secondly, evaluating the quality of augmented training data. Thirdly, the pre-training of the language model, and fourthly, the evaluation of the language models using zero-shot and fine-tuning tasks.

4.1 Data Generation

The experiments use the 10-million-word corpus from the official `strict-small` 2025 BabyLM Challenge by Charpentier et al. (2025). We denote the $\approx 10M$-word internal baseline dataset from Table 4 as $\mathcal{D}_{base}$, with $\mathcal{D}_{baseX}$ representing the sampled proportion of X million words; $\mathcal{D}_{augY}$ as the Y million words generated via RTA; and the combined dataset as $\mathcal{D}_{baseX} + \mathcal{D}_{augY}$. To create the different proportions from the baseline dataset $\mathcal{D}_{base}$, the sample chunks and split script from Warstadt et al. (2025) is used. The custom Python script from Timiryasov and Tastet (2023) is used for preprocessing (see Appendix A).

GloVe (Pennington et al., 2014) is employed to generate word embeddings and trained on the respective baseline proportion for each variant. Sentence embeddings are created corpus-wide with uSIF (Ethayarajh, 2018) when running the algorithm. RTA was applied to the baseline dataset $\mathcal{D}_{\text{base}}$ to generate synthetic sentences in $\mathcal{D}_{\text{aug}}$. The augmented sentences are inserted adjacent to the reference sentences, based on the findings from Haga et al. (2024).

4.2 Data Quality

Perplexity (PPL) is used to evaluate the quality of the generated data $\mathcal{D}_{\text{aug}}$. The evaluation of data quality is only a diagnostic metric and independent of the data generation process. The perplexity is compared for all augmented sentences in $\mathcal{D}_{\text{aug}}$ with the perplexity values for all reference sentences from $\mathcal{D}_{\text{base}}$ to determine whether the augmented examples preserve the underlying linguistic fluency. For this, the official pre-trained openai-community/gpt2 (Radford et al., 2019) from Huggingface is utilized, as it has been trained on large amounts of data, enabling us to determine how surprised the model is by the augmented sentences. The RTA-augmented datasets used in our experiments exhibit an approximately three times higher average PPL (53.09 ± 4.11) compared to the proportional baseline datasets (17.83 ± 0.21). Self-BLEU scores are used to measure the diversity within the corpus and show a modest increase for augmented datasets ($14.57\% \pm 4.45\%$) compared to the proportional baseline datasets ($8.41\% \pm 0.31\%$).

4.3 Language Model Pre-training

For language model pre-training, we utilize a decoder-based language model, GPT-2 Radford et al. (2019), which was trained using next-token prediction. Additionally, we employ an encoder-based language model, RoBERTa (Liu et al., 2019), which was trained using masked language modeling with a custom checkpoint strategy. Training is quantified by the total number of whitespace-separated input words, which must not exceed the threshold of 100 million input words in total for all models trained on text (Charpentier et al., 2025) (for details see Appendix A).

4.4 Language Model Evaluation

For the performance evaluation of the pre-trained language model, the official 2025 BabyLM Challenge evaluation pipeline from Charpentier et al. (2025) is utilized, which encompasses domain-specific zero-shot, fast zero-shot, and fine-tuning tasks.

For zero-shot evaluation, the tasks are the Benchmark of Linguistic Minimal Pairs (BLiMP) and the BLiMP Supplement (Warstadt et al., 2020), which assess grammatical ability. Elements of World Knowledge (EWoK) (Ivanova et al., 2024) evaluates the ability of world modeling for language models. Eye Tracking (Reading ET) and Self-paced Reading (Reading SPR) (de Varda et al., 2024) are behavioral paradigms used to measure word-by-word language processing. Entity Tracking (ENT) (Kim and Schuster, 2023; Charpentier et al., 2025) tests how well language models can follow changes to objects such as a book or an apple within a story or conversation. WUGs (Weissweiler

et al., 2023; Hofmann et al., 2025) tests the linguistic generalization of language models. COMPS (Misra et al., 2022) assesses commonsense property inheritance, where properties of broader categories apply to more specific subcategories. Age of Acquisition (AoA) (Chang and Bergen, 2022) tracks word surprisal over training checkpoints to derive learning curves correlated with human acquisition ages. The fast zero-shot tasks utilize a smaller set of evaluation examples from the zero-shot tasks and evaluate the performance of the language models at specific checkpoints. The fine-tuning tasks utilize selected tasks from the GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) datasets, which reflect language understanding tasks.

5 Results and Discussion

The quantitative results are presented briefly below, followed by a brief insight into the qualitative results.

Decoder For our GPT-2 model (Table 1), the dataset variant that uses only 25% of the original data (25/75) achieves the highest performance for entity tracking (Kim and Schuster, 2023; Charpentier et al., 2025). The variant with 50% of the original data (50/50) achieves the highest performance for BLiMP Supplement (Warstadt et al., 2020), EWoK (Ivanova et al., 2024), WUGs Past, Reading SPR and ET (de Varda et al., 2024), and GLUE (Wang et al., 2018, 2019) over the baseline. The 75% original data variant (75/25) achieves the best results for BLiMP (Warstadt et al., 2020). The baseline showed a negligible correlation with human language acquisition, while the augmented datasets led to moderate improvements. The variant (75/25) achieved the highest correlation. However, no variant achieved statistical significance of $p < 0.05$ (see Appendix A).

Encoder For our RoBERTa model (Table 2), the dataset variant that uses only 25% of the original data (25/75) achieves the highest average performance for entity tracking (Kim and Schuster, 2023; Charpentier et al., 2025), WUGs Past (Weissweiler et al., 2023; Hofmann et al., 2025), and COMPS (Misra et al., 2022) benchmarks over the baseline. The detailed fine-tuning results show that the mixed variants achieve improvements in reading comprehension (Khashabi et al., 2018), recognizing text entailments (Dagan et al., 2005; Giampiccolo et al., 2007; Bentivogli et al., 2009), and identifying the meaning of an ambiguous word (Levesque et al., 2012). Regarding the correlation with human language acquisition, the variants consistently showed near-zero correlations (see Appendix A for details).

We want to highlight that all our model variants pre-trained on augmented data saw fewer words in total (see Appendix A.3) but achieved similar or better results in the zero-shot and fine-tuning tasks.

Qualitative Insights Table 3 presents qualitative examples generated with the RTA method and possibly explains the pronounced gains in entity tracking and grammatical understanding of the LMs. For example, the sentence `"Two months passed, and spring deepened into summer"` and the augmented version `"Days deepened into summer"` could improve entity tracking (Kim and Schuster, 2023; Charpentier et al., 2025) by highlighting time progressions. Similarly, swapping phrases such as `"My Missionary life has, on the whole, been a very happy one..."` with `"My Missionary life has, on the whole, was very happy"` leads to syntactic robustness, which may have resulted in stronger BLiMP (Warstadt et al., 2020) results in the augmented datasets. Mixing weather descriptions, as in the examples `"It was very early on a hot December morning."` and `"On nice sunny days when it was not very cold she took them out in the carriage,"` could, for example, expand world knowledge and thus have led to improvements in EWoK (Ivanova et al., 2024) and reading tasks (de Varda et al., 2024).

6 Conclusion

RecombiText Augmentation is a statistical compositional data augmentation approach that generates synthetic sentences via lexical-semantic similarities and a one-point crossover. Our experimental results show that the proposed augmentation method can expand low-resource datasets by two- to fourfold without degrading language model benchmark performance, and on tasks such as entity tracking, self-paced reading, and morphological generalization, it even outperforms models trained on the full original dataset. This highlights substantial data efficiency gains over training solely on the full original dataset.

		BLiMP				WUGs			Reading		
Variant	**Prop.**	**BLiMP**	**Suppl.**	**EWoK**	**ENT**	**Adj.**	**Past**	**COMPS**	**SPR**	**ET**	**GLUE**
Baseline	0%	61.62	58.17	50.01	12.95	0.69	-0.06	**51.48**	4.01	11.83	64.02
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	10%	58.30	57.42	49.81	16.24	0.47	-0.17	50.38	3.92	10.07	64.06
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	25%	60.18	60.48	50.41	**19.65**	0.54	**0.24**	50.49	3.97	11.65	64.50
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	50%	61.87	**61.08**	**50.80**	16.51	0.56	**0.24**	50.83	**4.74**	**12.05**	**64.76**
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	75%	**62.91**	59.02	49.74	17.50	0.56	0.03	51.42	4.13	11.64	63.39
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	90%	61.27	59.80	50.24	16.53	**0.71**	-0.05	50.84	4.39	11.80	62.70
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	100%	61.99	59.11	50.20	16.19	0.54	0.07	51.22	4.32	12.02	64.66

Table 1: GPT-2 evaluation results with downsampled size from baseline dataset (prop.), Blimp, Bimp Supplement (Suppl.), EWoK, Entity Tracking (ENT), COMPS with accuracy (in %), WUGs with Spearman's rank correlation coefficient ρ, Self-paced Reading (SPR), Eye Tracking (ET) with change in R^2, and (Super)GLUE subset with macroaverage accuracy (in %).

		BLiMP				WUGs			Reading		
Variant	**Prop.**	**BLiMP**	**Suppl.**	**EWoK**	**ENT**	**Adj.**	**Past**	**COMPS**	**SPR**	**ET**	**GLUE**
Baseline	0%	54.44	50.98	**51.01**	32.27	0.58	-0.09	50.34	3.08	10.06	61.27
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	10%	**57.06**	52.33	50.17	28.41	0.52	-0.03	50.66	**3.53**	10.17	61.51
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	25%	55.15	51.77	49.38	**38.89**	0.48	**0.23**	**51.25**	3.52	11.01	61.19
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	50%	55.56	51.55	49.89	33.65	0.67	-0.03	50.93	3.50	**11.02**	**62.29**
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	75%	55.76	52.72	49.92	29.02	0.68	-0.20	50.90	3.42	**11.03**	61.65
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	90%	54.30	51.67	49.62	33.83	0.57	-0.15	50.85	2.93	10.25	62.11
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	100%	54.39	**53.10**	49.46	29.70	**0.72**	-0.03	50.49	2.81	10.48	61.98

Table 2: RoBERTa evaluation results with downsampled size from baseline dataset (prop.), Blimp, Bimp Supplement (Suppl.), EWoK, Entity Tracking (ENT), COMPS with accuracy (in %), WUGs with Spearman's rank correlation coefficient ρ, Self-paced Reading (SPR), Eye Tracking (ET) with change in R^2 and (Super)GLUE subset with macroaverage accuracy (in %).

Reference Sentences	**Augmented Sentences**
Two months passed, and spring deepened into summer.	Two months passed, and spring ran into weeks, weeks into months, but the expected agent of deliverance was not forthcoming.
Days ran into weeks, weeks into months, but the expected agent of deliverance was not forthcoming.	Days deepened into summer.
Life on the whole was very happy. My Missionary life has, on the whole, been a very happy one....'	Life on the whole, been a very happy one....' My Missionary life has, on the whole was very happy.
I've already started something. Cos I've been spectating	I've been spectating Cos I've already started something.
It was very early on a hot December morning.	it was not very cold she took them out in the carriage.
On nice sunny days when it was not very cold she took them out in the carriage.	On nice sunny days when It was very early on a hot December morning.

Table 3: Examples of RTA-generated augmented sentences based on a picked reference sentence and matching candidates.

Learnings

The experiments have demonstrated that the proposed method improves the morphological generalization, entity tracking, and reading comprehension capabilities of language models over the baseline. However, the strategy used, the frequency with which query sentences are combined with candidate sentences, and how augmentation is performed

play a significant role. Based on the experiments and results, it can be assumed that language models develop a better understanding of language when they frequently see how the same sentence can be combined in different ways. This can be observed, for example, in variants with 25% original data (25/75), which yield better results in entity tracking and morphological generalization than more balanced variants.

Limitations

The success of semantic search for candidates and the context window depends on word embeddings. As a result, ineffective or low-quality embeddings can lead to candidates that are less semantically relevant and therefore influence the relevance for the resulting augmented sentences. Furthermore, as the embedding quality decreases, the algorithm relies more heavily on lexical matches within the context windows for the respective intersections. It is plausible to assume that such factors could influence the performance when evaluating the language models.

Focusing on the evaluation of the quality of word and sentence embeddings when using the RTA method could ensure robust semantic alignment. Furthermore, a separate assessment of the effects of lexical versus semantic augmentations can lead to a better understanding of the respective improvements. The selection of the matching candidates and the presentation of the resulting data to the language model could provide deeper insights into the individual contributions. This separate evaluation would further contribute to the optimization of the RTA method by showing which type of augmentation leads to improvements. Investigating the effects of the used RTA hyperparameters could also enable more targeted data augmentation strategies for low-resource scenarios.

References

Jacob Andreas. 2020. Good-Enough Compositional Data Augmentation.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, and 1 others. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *arXiv preprint arXiv:2502.10645*.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. Gpt or bert: why not both? *arXiv preprint arXiv:2410.24159*.

BNC Consortium and 1 others. 2007. British national corpus. *Oxford Text Archive Core Collection*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Lukas Edman, Lisa Bylinina, Faeze Ghorbanpour, and Alexander Fraser. 2024. Are BabyLMs Second Language Learners? *arXiv preprint*. ArXiv:2410.21254 [cs].

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

Kawin Ethayarajh. 2018. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 91–100.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society.

David E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edition. Addison-Wesley Longman Publishing Co., Inc., USA.

Akari Haga, Akiyo Fukatsu, Miyu Oba, Arianna Bisazza, and Yohei Oseki. 2024. BabyLM Challenge: Exploring the Effect of Variation Sets on Language Model Training Efficiency. *arXiv preprint*. ArXiv:2411.09587 [cs].

Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2020. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv preprint arXiv:2010.12309*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Valentin Hofmann, Leonie Weissweiler, David R Mortensen, Hinrich Schütze, and Janet B Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Michael Y Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. *arXiv preprint arXiv:2412.05149*.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, and 1 others. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Daniel Jurafsky and James H Martin. 2019. Speech and language processing 3rd edition draft. *October 2019*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. *arXiv preprint arXiv:2305.02363*.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. *KR*, 2012(13th):3.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Alex Lyman and Bryce Hepner. 2024. Whatif: Leveraging word vectors for small-scale data augmentation. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 229–236.

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Kanishka Misra, Julia Taylor Rayz, and Allyson Ettinger. 2022. Comps: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. *arXiv preprint arXiv:2210.01963*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389. Publisher: Now Publishers.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Ole Tange. 2025. Gnu parallel 20250522 ('leif tange'). GNU Parallel is a general parallelizer to run multiple serial command line programs in parallel without changing them.

Nikitas Theodoropoulos, Giorgos Filandrianos, Vassilis Lyberatos, Maria Lymperaiou, and Giorgos Stamou. 2024. BERTtime Stories: Investigating the Role of Synthetic Story Data in Language pre-training. *arXiv preprint*. ArXiv:2410.15365 [cs].

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. *arXiv preprint arXiv:2308.02019*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Warstadt and Samuel R Bowman. 2022. What artificial neural networks can tell us about human language acquisition. In *Algebraic structures in natural language*, pages 17–60. CRC Press.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and 1 others. 2025. Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora. *arXiv preprint arXiv:2504.08165*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, and 1 others. 2023. Counting the bugs in chatgpt's wugs: A multilingual investigation into the morphological capabilities of a large language model. *arXiv preprint arXiv:2310.15113*.

Zheyu Zhang, Han Yang, Bolei Ma, David Rügamer, and Ercong Nie. 2023. Baby's CoThought: Leveraging Large Language Models for Enhanced Reasoning in Compact Models. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 158–170, Singapore. Association for Computational Linguistics.

A Appendix A

A.1 Setup Details

The RTA algorithm was implemented locally with Python 3.13. GNU Parallel (Tange, 2025) was used to execute the jobs for generating the augmented datasets, enabling multiple instances to be run simultaneously. The language models were trained on 1x NVIDIA H100 Tensor Core GPU with Python 3.12.9, Transformers 4.50.3, and PyTorch 2.7.1+cu126. For reproducibility, random seeds are used.

A.2 Data Preprocessing

The datasets were preprocessed to remove specific metadata. The custom Python script from Timiryasov and Tastet (2023) was used. We apply corpus-specific cleanup functions to normalize the text for model training. The common steps for all functions include removing extra spaces, tabs, and unnecessary line breaks. Additional customized processes vary depending on the corpus. For example, in OpenSubtitles (Lison and Tiedemann, 2016), subtitle credits are removed. In Simple English Wikipedia, leading line breaks at the end are removed.

A.3 Language Model Pre-training Details

GloVe (Pennington et al., 2014) is trained exclusively on the baseline proportion $|\mathcal{D}_{baseX}|$, which is deducted from the total 100 million word budget to determine the remaining allocation for LM training. The training corpus was tokenized using Byte Pair Encoding (BPE) (Gage, 1994; Sennrich et al., 2015) with the script from Charpentier and Samuel (2024). This yields the average number of subword tokens per whitespace-separated word (Avg. Splits/Word), which varies slightly across variants due to differences in data composition. All LMs use a batch size of 16 and a sequence length of 512, resulting in $16 \times 512 = 8192$ tokens per training step. The approximate number of words processed per step is then calculated as $\text{Words/Step} = \frac{\text{Tokens/Step}}{\text{Avg. Splits/Word}}$. To utilize the allocated number of words for the LMs without exceeding the budget, the maximum training steps are determined by the formula $\text{Max Steps} = \frac{\text{\# Words LM}}{\text{Words/Step}}$. The number of times the LMs iterate over the dataset is called an epoch and is, in our case, calculated as $\text{Epoch} = \frac{\text{\# Words LM}}{\text{\# Words Dataset}}$. Checkpoints are created every 1 million words for the first 10 million words and every 10 million words thereafter, up to a total of 100 million words. To determine when the language models have encountered a specific number of words, we perform specific calculations. Table 6 shows the detailed experimental language model setup for each variant.

Dataset	Description	Citation	# Words
British National Corpus (BNC)	Dialogue	(Consortium et al., 2007)	0.93M
CHILDES	Child-directed speech	(MacWhinney, 2000)	2.84M
Project Gutenberg (children's stories)	Written English	(Gerlach and Font-Clos, 2020)	2.54M
OpenSubtitles	Movie subtitles	(Lison and Tiedemann, 2016)	2.04M
Simple English Wikipedia	Written Simple English	-	1.45M
Switchboard Dialog Act Corpus	Dialogue	(Godfrey et al., 1992; Stolcke et al., 2000)	0.15M
Total			9.95M

Table 4: Datasets for the experiments simulating the low-resource scenario by Charpentier et al. (2025) after preprocessing.

Variant	Proportion	# Total Words	# Base Words	# Aug Words	Ratio r
$\mathcal{D}_{base10M}$	0%	9.95M	9.95M	0.00M	0.00
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	10%	10.55M	1.06M	9.49M	9.00
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	25%	9.68M	2.43M	7.25M	3.00
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	50%	9.72M	4.96M	4.76M	1.00
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	75%	9.99M	7.48M	2.52M	0.33
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	90%	9.96M	8.95M	1.01M	0.11
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	100%	19.49M	9.95M	9.54M	1.00

Table 5: Sampled proportions from the original `strict-small` low-resource training data by Charpentier et al. (2025), total words (in millions) in the resulting dataset, actual number of base words sampled from the baseline dataset $\mathcal{D}_{base}$, augmented words generated as $\mathcal{D}_{aug}$ with RTA and the approximate nominal augmentation ratio $r = \frac{|\mathcal{D}_{aug}|}{|\mathcal{D}_{base}|}$. The actual ratios may vary slightly due to data preprocessing.

Variant	# Words Dataset	Avg. Splits/Word	# Words GloVe	# Words LM	Tokens/Step	Words/Step	Max Steps	Epochs
$\mathcal{D}_{base10M}$	9.95M	1.608	0.00M	100.00M	8,192	5,095	19,627	10.05
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	10.55M	1.463	1.06M	98.94M	8,192	5,599	17,672	9.38
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	9.68M	1.518	2.43M	97.57M	8,192	5,397	18,079	10.08
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	9.72M	1.529	4.96M	95.04M	8,192	5,358	17,738	9.77
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	9.99M	1.564	7.48M	92.52M	8,192	5,238	17,664	9.26
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	9.96M	1.59	8.95M	91.05M	8,192	5,152	17,672	9.14
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	19.49M	1.527	9.95M	90.05M	8,192	5,365	16,785	4.62

Table 6: Language Model training setup for each data augmentation variant, where # Words represents the total whitespace-separated words in the preprocessed training dataset, Avg. Splits/Word indicates the average subword tokens per word from BPE tokenizer, # Words GloVe is the size of the dataset with which the GloVe model was trained, # Words LM reflects the budgeted words for the language model, Tokens/Step is the number of tokens per training step the LM sees, Words/Step approximates the number of words per training step the LM sees, Max Steps gives the total training steps for the LM, Epochs is the number the LM iterates over the dataset.

Variant	PPL		Self-BLEU	
	D_{baseX}	D_{augY}	D_{baseX}	D_{augY}
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	18.23	47.50	8.83	12.65
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	17.69	49.84	8.67	12.56
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	17.83	54.21	8.50	12.33
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	17.81	55.05	8.02	18.33
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	17.66	59.21	8.30	22.67
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	17.75	53.51	8.19	11.74

Table 7: Data quality evaluation for the base dataset $\mathcal{D}_{base}$ and the augmented dataset $\mathcal{D}_{aug}$ with their respective PPL and Self-BLEU (in %) values.

Variant	AoA	
	GPT-2	RoBERTa
Baseline	-0.001 (p=0.997)	0.00
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	0.120 (p=0.670)	-0.274 (p=0.656)
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	0.233 (p=0.404)	0.00
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	0.031 (p=0.888)	0.00
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	0.265 (p=0.182)	0.00
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	0.089 (p=0.718)	0.00
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	0.204 (p=0.388)	0.00

Table 8: AoA results for the different model variants.

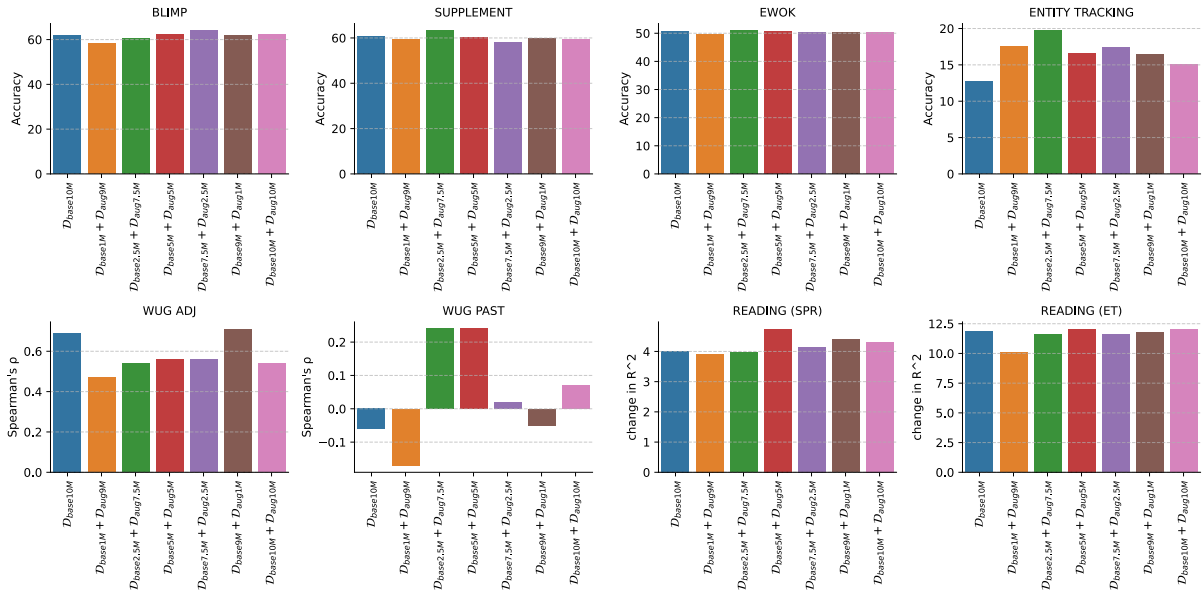


Figure 2: GPT-2 final fast zero-shot checkpoint results.

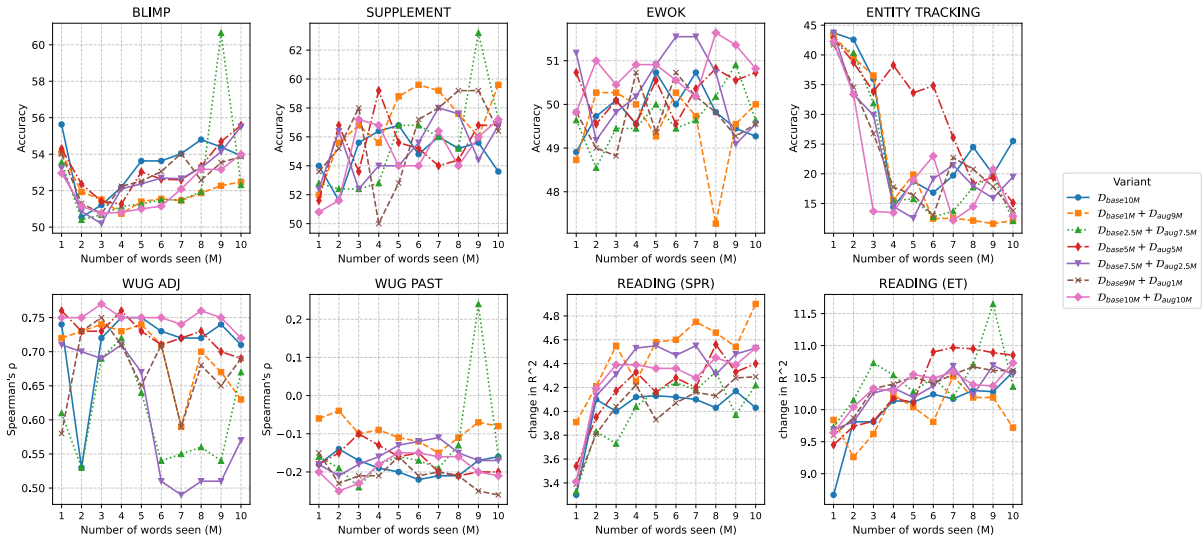


Figure 3: GPT-2 fast zero-shot results across training checkpoints where the language model has seen between 1M–10M words.

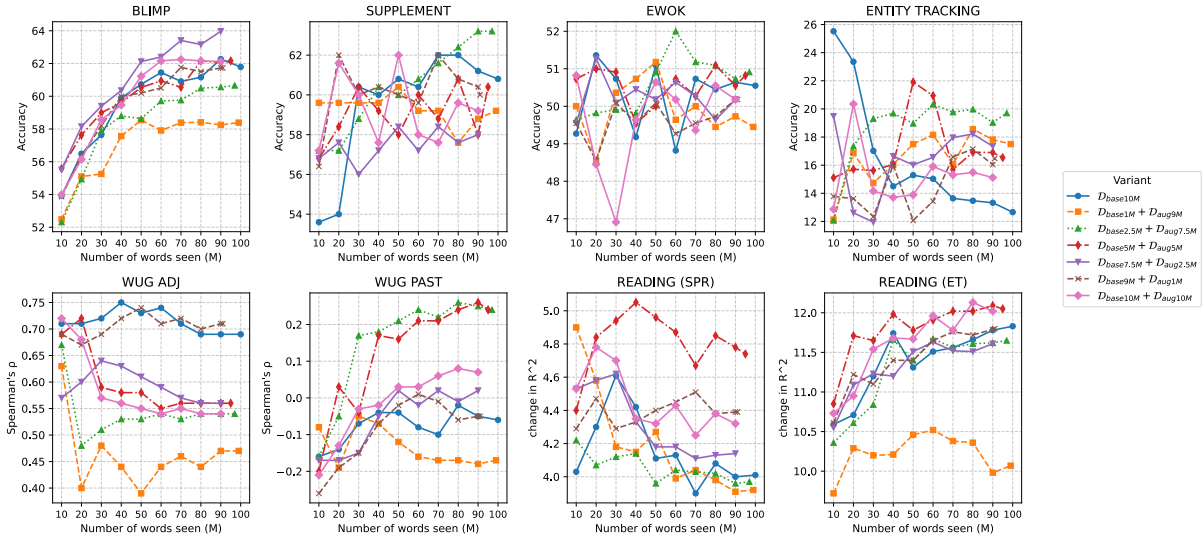


Figure 4: GPT-2 fast zero-shot results across training checkpoints where the language model has seen between 10M–100M words.

Variant	Prop.	BoolQ	MNLI	MRPC	QQP	MultiRC	RTE	WSC	Macro Avg.
Baseline	0%	67.58	50.81	81.25	62.86	**63.94**	**60.43**	**67.31**	64.02
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	10%	67.65	54.79	82.46	66.99	63.94	53.96	63.46	64.06
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	25%	67.03	54.16	82.01	66.97	63.94	59.71	63.46	64.50
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	50%	66.73	**55.89**	**83.23**	66.17	60.81	59.71	65.38	**64.76**
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	75%	66.91	52.89	81.96	66.73	59.98	56.12	65.38	63.39
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	90%	67.83	49.29	81.37	61.22	60.35	58.27	63.46	62.70
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	100%	**68.75**	55.62	82.65	**68.20**	60.60	56.12	65.38	64.66

Table 9: GPT-2 detailed fine-tuning results.

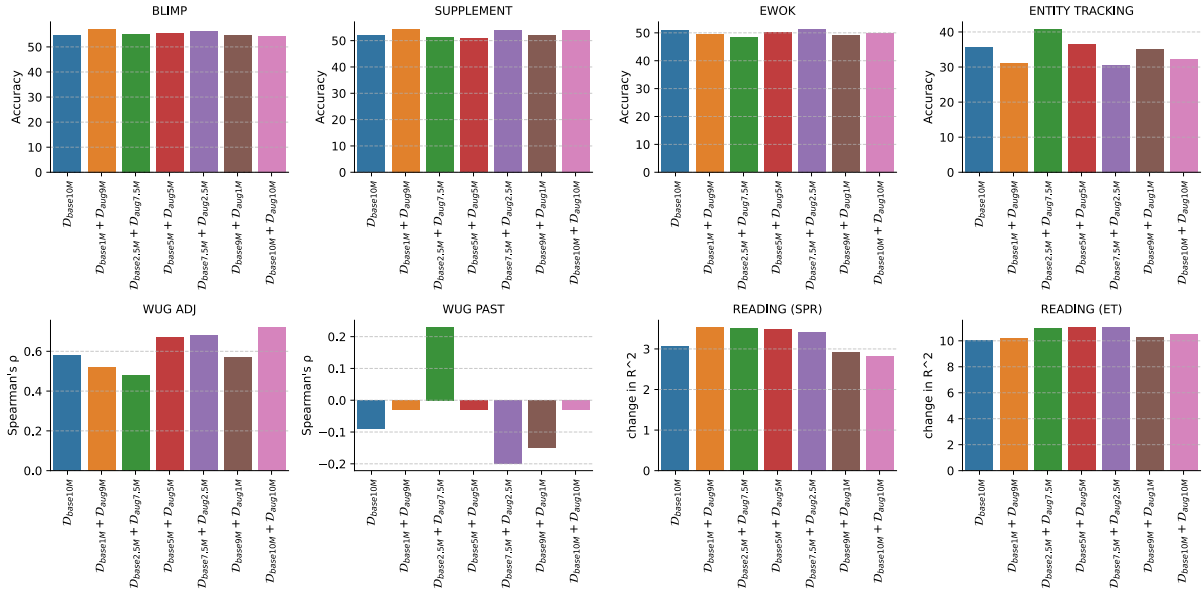


Figure 5: RoBERTa final fast zero-shot checkpoint results.

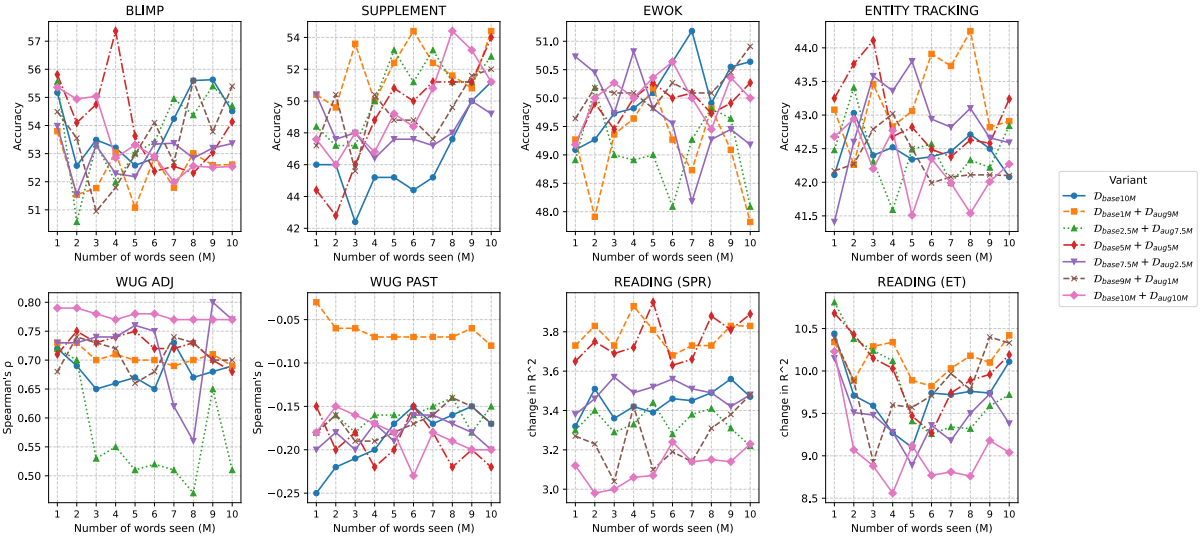


Figure 6: RoBERTa fast zero-shot results across training checkpoints where the language model has seen between 1M–10M words.

Variant	Prop.	BoolQ	MNLI	MRPC	QQP	MultiRC	RTE	WSC	Macro Avg.
Baseline	0%	66.24	40.99	**83.39**	59.46	57.55	54.68	65.38	61.27
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	10%	66.36	42.32	81.23	**61.18**	57.96	56.12	65.38	61.51
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	25%	66.12	41.95	81.55	60.33	57.55	56.83	61.54	61.19
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	50%	66.54	**43.42**	82.47	61.08	**60.23**	55.40	65.38	**62.29**
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	75%	**67.22**	43.13	82.28	60.31	57.96	56.83	65.38	61.65
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	90%	66.67	42.46	82.13	60.59	57.96	**58.99**	**67.31**	62.11
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	100%	65.87	42.14	80.75	58.47	59.32	57.55	**67.31**	61.98

Table 10: RoBERTa detailed fine-tuning results.

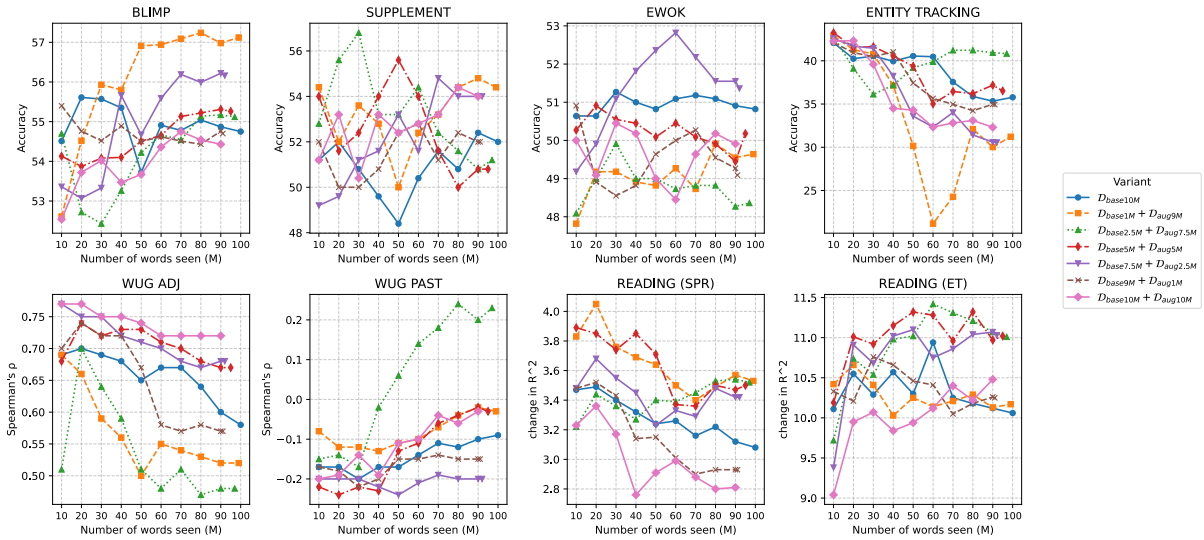


Figure 7: RoBERTa fast zero-shot results across training checkpoints where the language model has seen between 10M–100M words.

Hyperparameter	**Value**
Vector Size	50
Window Size	10
Minimum Vocabulary Count	5
Maximum Co-occurrence Weight (`x_max`)	10
Maximum Iterations	25
Number of Threads	4
Memory	4.0 GB

Table 11: Hyperparameters for the GloVe model used in the experiments.

Variant	**RRF** k	τ_{window}	W	**top-**k	T	**Ratio** r
$\mathcal{D}_{base1M} + \mathcal{D}_{aug9M}$	60	60%	3	50	1.3	9.00
$\mathcal{D}_{base2.5M} + \mathcal{D}_{aug7.5M}$	60	60%	3	30	1.1	3.00
$\mathcal{D}_{base5M} + \mathcal{D}_{aug5M}$	60	60%	3	20	1.0	1.00
$\mathcal{D}_{base7.5M} + \mathcal{D}_{aug2.5M}$	60	60%	3	15	0.9	0.33
$\mathcal{D}_{base9M} + \mathcal{D}_{aug1M}$	60	60%	3	10	0.8	0.11
$\mathcal{D}_{base10M} + \mathcal{D}_{aug10M}$	60	60%	3	20	1.0	1.00

Table 12: Hyperparameters for the RTA algorithm used in the experiments.

Hyperparameter	Value
Model Type	openai-community/gpt2
Tokenizer	BPE
Learning Rate	5×10^{-5}
Maximum Gradient Norm	1.0
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	1×10^{-8}
Block Size	512
Batch Size	16
Save Strategy	Steps
Save Total Limit	20
Logging Steps	100
Evaluation Strategy	Steps
Seed	42

Table 13: Hyperparameters for the GPT-2 model training used in the experiments.

Hyperparameter	Value
Model Type	FacebookAI/roberta-base
Tokenizer	BPE
Learning Rate	5×10^{-5}
Maximum Gradient Norm	1.0
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	1×10^{-8}
Sequence Length	512
Batch Size	16
MLM Probability	15%
Save Strategy	Steps
Save Total Limit	20
Logging Steps	100
Evaluation Strategy	Steps
Seed	42

Table 14: Hyperparameters for the RoBERTa model training used in the experiments.

Hyperparameter	MultiNLI, RTE, QQP, MRPC	BoolQ, MultiRC	WSC
Learning Rate	3×10^{-5}	3×10^{-5}	3×10^{-5}
Batch Size	32	16	32
Epochs	10	10	30
Weight Decay	0.01	0.01	0.01
Optimizer	AdamW	AdamW	AdamW
Scheduler	cosine	cosine	cosine
Warmup Percentage	6%	6%	6%
Dropout	0.1	0.1	0.1

Table 15: Hyperparameters for fine-tuning the language models used in the experiments.

Expression	Definition
D	Set of documents
q	Query sentence
$\|q\|, \|m\|$	Sentence lengths
$\tilde{q}, \tilde{m}$	Augmented sentence pairs
$\mathcal{R}_{BM25}$	Lexical search results (BM25)
$\mathcal{R}_{kNN}$	Semantic search results (k-NN)
$\mathcal{R}_{RRF}$	Fused ranked list from Reciprocal Rank Fusion
τ_{window}	Threshold for context window similarity
k_{top}	Number of top-k candidates for sampling
$p(x_i)$	Softmax probability
m	Candidate sentence
W	Fixed size of the context window
$\mathbf{v}_t \in \mathbb{R}^d$	Word embedding of token t in d-dimensional space
$\mathbf{v}_{uSIF} \in \mathbb{R}^d$	Sentence embeddings in d-dimensional space
$\text{idf}(t, D)$	IDF value of token t in corpus D
$S_{\text{ctx}}(i, j)$	Context window similarity
i, j	Positions within the sliding context windows
k^*	Pivot index within the context window maximizing similarity
i^*, j^*	Pivot elements for q, m
$\|\|$	Concatenation operator

Table 16: Mathematical notation for the RecombiText Augmentation (RTA) algorithm.

Algorithm 1: RecombiText Augmentation (RTA) Pseudo Algorithm

Data: Dataset D, $q \in D$, $\mathbf{v}_t \in \mathbb{R}^d$, $\mathbf{v}_{uSIF} \in \mathbb{R}^d$, $\mathrm{idf}(t, D)$
Result: $\tilde{q}, \tilde{m}$
Retrieve $\mathcal{R}_{BM25}$ for q;
Retrieve $\mathcal{R}_{kNN}$ for q on $\mathbf{v}_{uSIF} \in \mathbb{R}^d$;
$\mathcal{R}_{RRF} \leftarrow empty$;
for d *in* $\mathcal{R}_{BM25} \cup \mathcal{R}_{kNN}$ **do**
    Compute RRFscore;
    Add d to $\mathcal{R}_{RRF}$ with its RRF score;
end
Sort $\mathcal{R}_{RRF}$ in descending order of RRF scores;
$retry \leftarrow 1$;
for *retry* $\leq |\mathcal{R}_{RRF}|$ **do**
    Select k_{top} candidates from $\mathcal{R}_{RRF}$;
    Compute $p(x_i)$ on k_{top} candidates according to top-k sampling;
    Sample candidate sentence m according to probabilities p;
    Tokenize $q = \langle q_0, \ldots, q_{|q|-1} \rangle$ and $m = \langle m_0, \ldots, m_{|m|-1} \rangle$;
    Normalize q, m;
    $S_{\max} \leftarrow 0$, $i^* \leftarrow 0$, $j^* = 0$;
    for $i = 0$ **to** $|q| - W_q + 1$ **do**
        for $j = 0$ **to** $|m| - W_m + 1$ **do**
            Compute $S_{\text{ctx}}(i, j)$ according to Equation 1;
            if $S_{ctx}(i, j) > S_{\max}$ **then**
                $S_{\max} \leftarrow S_{\text{ctx}}(i, j)$;
                Compute k^* for W_q, W_m according to $\cos_k \times \mathrm{idf}_k$;
                $i^* \leftarrow i + k^*$;
                $j^* \leftarrow j + k^*$;
            end
        end
    end
    if $S_{\max} > \tau_{window}$ **then**
        $\tilde{q} = \langle q_0, \ldots, q_{i^*-1} \mid\mid m_{j^*}, \ldots, m_{|m|-1} \rangle$;
        $\tilde{m} = \langle m_0, \ldots, m_{j^*-1} \mid\mid q_{i^*}, \ldots, q_{|q|-1} \rangle$;
        return $\tilde{q}, \tilde{m}$;
    else
        $retry \leftarrow retry + 1$;
        Remove candidate m from $\mathcal{R}_{RRF}$;
        continue;
    end
end

Author Index