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 488–495

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

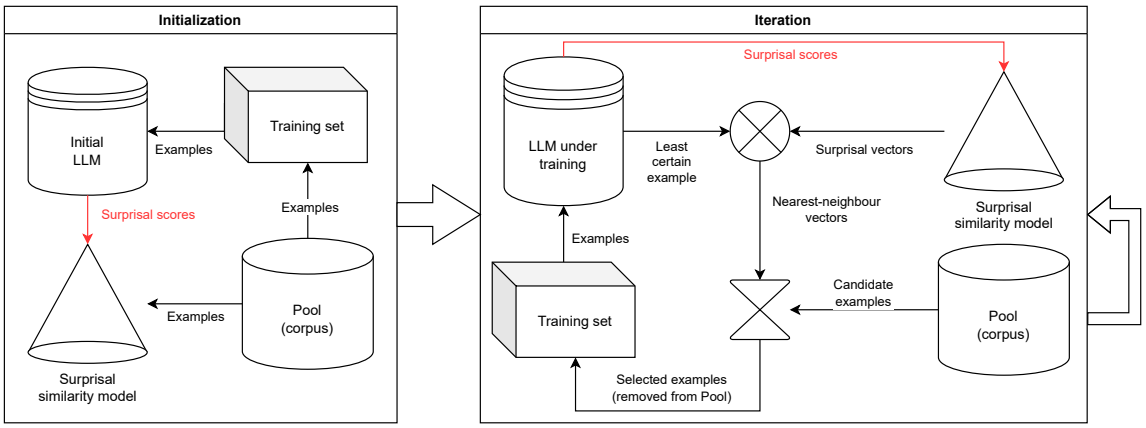

Figure 1: The architecture of our ACLM method from last year's submission, described in Hong et al. (2024).

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.