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,

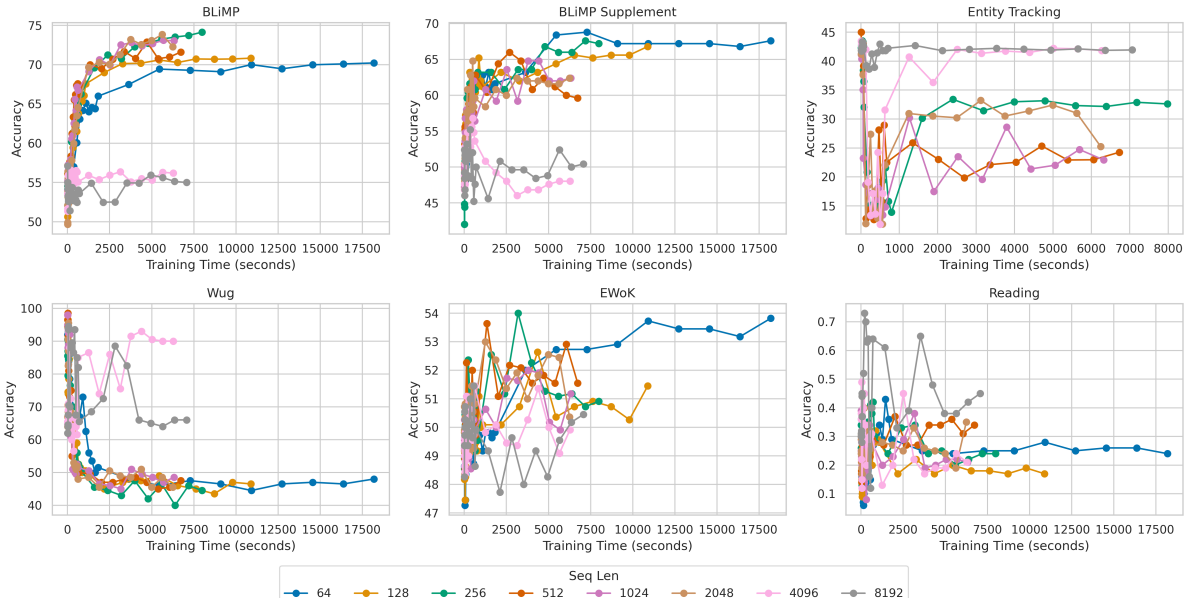


Figure 1: OPT Model Families: Effect of Sequence Length Accuracy vs Training Time per Metric. Evaluation of OPT 125M Family trained on 100M STRICT BabyLM Corpus with Warmup with a range of sequence lengths $\{64, 128, 256, 512, 1024, 2048, 8192\}$ on the Zero-Shot Evaluation Tasks of the 2025 BabyLM 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-

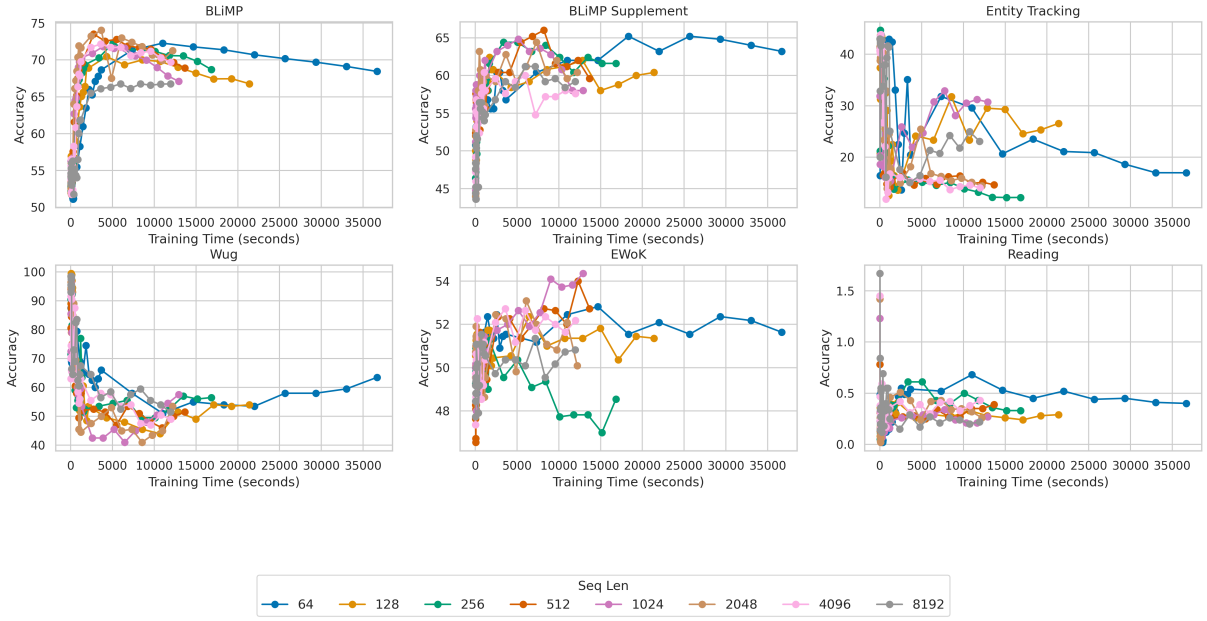


Figure 2: Mamba Model Families: Effect of Sequence Length Accuracy vs Training Time per Metric

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

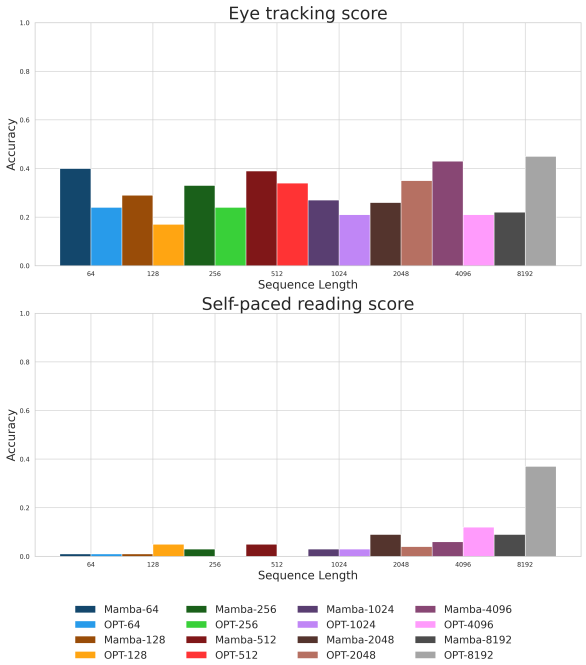


Figure 3: Distribution of Reading Sequence Length Model Accuracies for OPT Architecture

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)

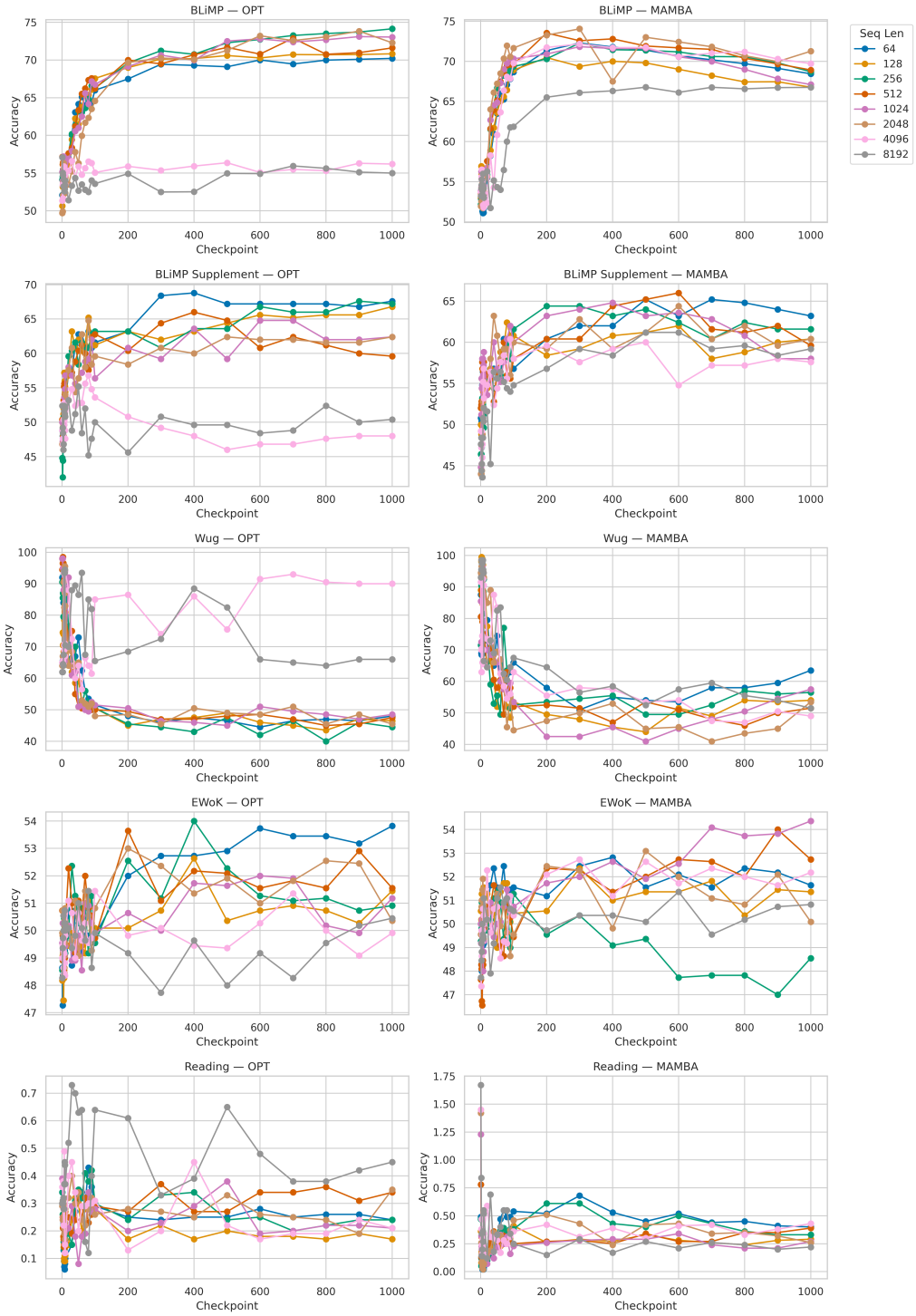
Table 6 provides a detailed breakdown of model performance on the full zero-shot evaluation tasks. In particular, we report differences between training models with and without warmup.

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

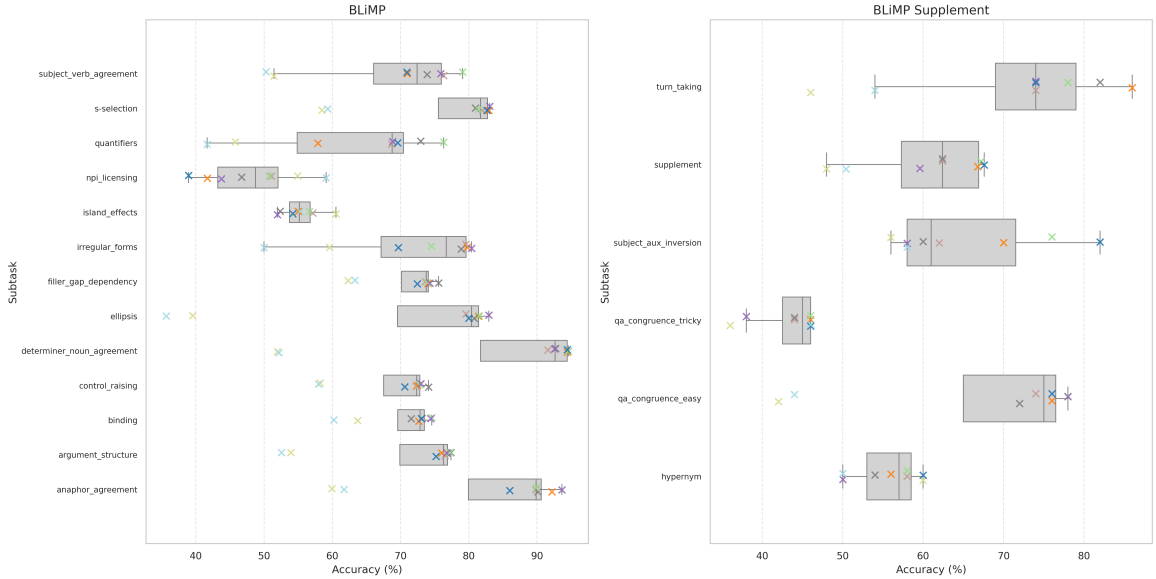


Figure 5: Distribution of OPT Sequence Length Model Accuracies on BLiMP and BLiMP Supplement

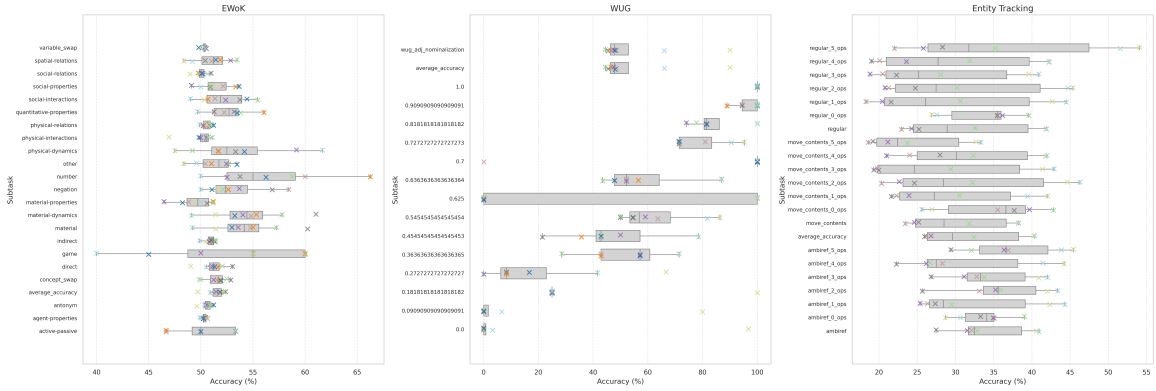


Figure 6: Distribution of EWoK, Wug and Entity Tracking Sequence Length Model Accuracies for OPT Architecture

F F1 Scores for Fine-Tuning

Figure 7: **F1 for Fine-Tuned Models**

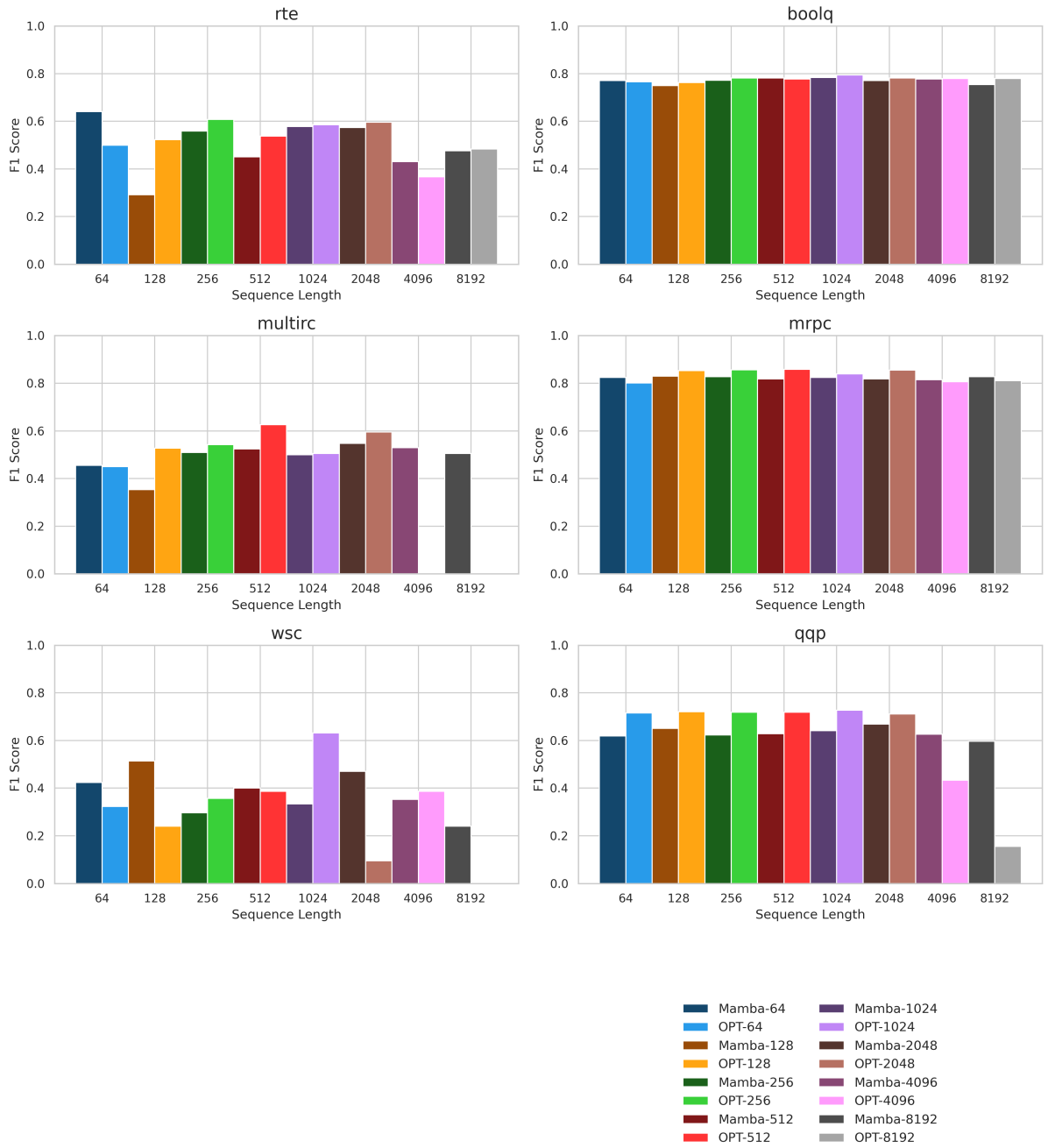


Figure 8: F1 Scores for OPT and Mamba Families on Fine-Tuned Tasks