A PAUSE IN TIME SAVES NINE: PUNCTUATION-GUIDED ATTENTION FOR EFFICIENT MULTI-TOKEN REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard self-attention scores each query against individual keys, which makes it difficult for Transformers to condition on the joint presence of multiple non-contiguous concepts. Prior remedies rely on fixed-size local neighborhoods that are not aligned with the semantic structure of text and scale poorly to long contexts. We introduce Punctuation-Guided Attention (PGA), which inserts learned summary tokens at punctuation boundaries (PAST) and restricts attention beyond a local sliding window to these summaries. This hierarchical pattern gives every token access to compressed representations of all preceding segments while filtering token-level noise. On a synthetic multi-token retrieval task, PGA maintains above 96% accuracy as block count and query length grow, where standard and summary-token-only Transformers collapse to near-zero accuracy. On real data, PGA improves convergence when pretraining from scratch and improves downstream accuracy under continued pretraining of Llama 3.2 1B. Because past standard tokens can be evicted once summarized, PGA also substantially reduces the inference-time key-value (KV) cache.

1 INTRODUCTION

The recent paradigm shift in artificial intelligence has been defined by the success of Large Language Models (LLMs). These models, built upon the Transformer architecture and its core self-attention mechanism (Vaswani et al., 2017), have demonstrated an unprecedented capacity for knowledge synthesis and complex reasoning. This success is rooted in a surprisingly simple principle: training for next-token prediction with a standard cross-entropy loss over massive datasets.

While the ability of self-attention to maintain uncompressed, global context was a key advantage over bottleneck-prone recurrent architectures (Hochreiter & Schmidhuber, 1997), this design choice introduces a fundamental trade-off. The mechanism's core operation, where each query token attends to key tokens independently, creates a significant impediment to multi-token reasoning (Golovneva et al., 2025; Kuratov et al., 2024). Specifically, the architecture is not naturally suited for tasks requiring the model to condition on the co-occurrence of multiple, non-contiguous concepts. This makes complex logical checks, such as verifying the presence of both "concept A" and "concept B" in a single context, a notoriously difficult challenge for standard Transformers (Liu et al., 2023).

In this work, we address this limitation with Punctuation-Guided Attention (PGA). PGA inserts special "summary tokens" (Mu et al., 2024; Goyal et al., 2024) at semantically meaningful locations, such as at the end of clauses or sentences (Razzhigaev et al., 2025). These tokens act as dedicated hubs for information aggregation, creating condensed representations of token groups. By attending to these summary tokens, the model can efficiently perform the complex, multi-token conditional reasoning that is difficult for standard Transformers.

The benefits of this architecture extend beyond training to inference efficiency. During autoregressive decoding, the summary tokens serve as natural waypoints for a sparse attention mechanism. Instead of attending to every token in the context, the model can focus primarily on the dense semantic information captured in the much smaller set of summary tokens. This allows for a highly compressed KV cache: by retaining only a local window plus the KVs of summary tokens, we re-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

duce the memory footprint by approximately $4\times$ even under dense punctuation, and by more on sparsely punctuated text (Section 3.3). Critically, this is not just an efficiency gain: by filtering out low-signal tokens, this focused attention also enhances the model's reasoning capabilities.

Our contributions are threefold:

- **Multi-token reasoning via Punctuation-based Aggregation via Summary Tokens (PAST).** We introduce a mechanism using summary tokens, placed at punctuation-defined boundaries, that provides dedicated capacity for aggregating information across multiple distinct concepts within a context (Section 3.1).
- **Scalable, noise-resistant attention via Punctuation-Guided Attention (PGA).** Our punctuation-guided attention mask provides a focused context, reducing noise from irrelevant tokens and scaling gracefully with longer sequences and harder queries (Section 3.2).
- **Inference efficiency via KV cache reduction.** Because standard tokens outside the local window can be evicted once summarized, PGA substantially reduces the KV cache required during decoding (Section 3.3).

2 Background and Motivation

In this section, we motivate our work by first examining the fundamental limitations of the standard self-attention mechanism in Transformers. We highlight two primary challenges that our approach aims to solve: (1) the difficulty in modeling complex dependencies that span multiple, non-contiguous tokens, and (2) the computational inefficiency and susceptibility to noise that arises from attending to an unfiltered history of all previous tokens. We then survey prior work that has sought to address these issues, from methods that enhance multi-token reasoning to those that focus on creating efficient, sparse attention patterns. This analysis situates our approach (PGA and PAST) within the existing research landscape, clarifying how it builds upon previous insights while offering a more general and scalable solution.

2.1 Limitations of Standard Self-Attention

A primary limitation of standard self-attention lies in its difficulty with complex tasks that require conditioning on multiple, distinct pieces of information. This challenge arises because attention scores are computed based on the similarity between a single query vector and a single key vector, making it difficult to model dependencies that are conditional on the presence of several concepts simultaneously. Prior work has attempted to address this; for example, Multi-Token Attention (Golovneva et al., 2025) employs convolutions to allow attention weights to be influenced by a neighborhood of tokens rather than just one. However, this approach is fundamentally limited by its reliance on an arbitrarily-sized local neighborhood, which is not aligned with the natural semantic structure of the text and does not scale efficiently to aggregation across longer non-contiguous contexts.

A second, related challenge is the inefficiency and noise introduced by providing the model with unrestricted access to the entire context history. The model must expend considerable capacity to determine which of the thousands of previous tokens are relevant and which are merely distractions. This "lost-in-the-middle" phenomenon has been well documented, with studies showing that models often ignore the vast majority of tokens in their context, particularly those in the middle of long sequences (Liu et al., 2023). Furthermore, research on the KV cache has empirically confirmed this inefficiency. Multiple studies have demonstrated that a significant fraction of cached tokens can be pruned during inference with little to no impact on accuracy, indicating that they do not contribute meaningfully to the final output (Jiang et al., 2024; Lai et al., 2025). Together, these issues highlight a clear need for an attention mechanism that can intelligently filter context and focus on semantically rich information.

2.2 Related Work

A primary line of related work focuses on improving reasoning by moving beyond the standard single-token attention mechanism. This body of work aims to condition attention scores on groups

of tokens simultaneously, providing a richer and more nuanced basis for locating relevant context. Variants of this concept have been successfully employed in sequence-to-sequence tasks (Wu et al., 2019) and speech processing (Gulati et al., 2020) to capture local neighborhood information. Methods such as Multi-Token Attention (Golovneva et al., 2025) have explored this for language but typically rely on fixed-size, local neighborhoods to aggregate information. This approach, while beneficial, is not inherently aligned with the variable-length semantic structure of text and faces challenges in scaling efficiently, leaving a need for a more semantically aware and scalable solution.

Another significant line of research has focused on reducing the noise and computational cost of self-attention, particularly by optimizing the KV cache. This is a rich field of literature with diverse strategies. Some methods identify and retain only the most salient columns of the attention matrix (Zhu et al., 2024). Other popular techniques include using low-rank approximations to project keys and values into a more compact subspace (Song et al., 2024), or introducing gating mechanisms to dynamically control information flow (Gao et al., 2025), among others. The prior work closest to our proposal is Activation Beacon (Zhang et al., 2024), which also introduces extra tokens to capture summaries at regular intervals, providing the model extra space for storing summaries as well as KV size reduction at inference time. While these efforts are effective, they are motivated by observing models' behavior and enhancing it. In contrast, our approach is fundamentally data-driven: instead of reacting to learned model states or fixed intervals, PGA preemptively leverages the inherent semantic structure of the input data itself to guide its attention strategy from the outset.

Our work shares conceptual similarities with a recurring theme in the literature: providing a model with extra computational space in the form of "thinking" tokens or a "scratchpad" (Goyal et al., 2024). These methods overlap with our PAST mechanism in that they also insert non-content tokens into the sequence to aid reasoning. However, a key distinction lies in the motivation and mechanism. Whereas this prior work often introduces tokens as a generic resource for unstructured reasoning or planning, our summary tokens are explicitly designed to capture a compressed representation of past semantic chunks, with their placement guided by the inherent structure of the input data. This distinction also points to a promising research direction: extending our framework so that these tokens not only summarize previous content but also serve as a dedicated space to plan future output, potentially merging the benefits of both approaches.

3 Punctuation-Guided Attention

This section describes our approach (shown in Figure 1) in three parts. First, we introduce our tokenization strategy, Punctuation-based Aggregation via Summary Tokens (PAST), which creates aggregation points at semantically meaningful breaks in the text. Next, we detail the hierarchical attention mechanism of PGA, which encourages the model to leverage these points for storing rich semantic information and performing complex reasoning. Finally, we analyze how this design is not only more capable but also significantly more deployment-friendly, discussing its impact on inference efficiency.

3.1 PAST: Semantic Segmentation with Summary Tokens

Recent work analyzing the internal mechanisms of LLMs has shown that certain tokens, particularly punctuation, often serve as crucial aggregation points for storing contextual information (Razzhigaev et al., 2025). Our approach leans into this observed behavior by explicitly augmenting the model's capacity at these natural semantic boundaries. Specifically, at punctuation marks within the input sequence, we segment the text and introduce N special summary tokens (N=4 in our real-data experiments). These tokens present the model with space to store a compressed representation of the preceding semantic chunk (i.e., the clause or sentence), providing dedicated space for the model to store richer context.

As illustrated in Figure 1, summary tokens are inserted into the sequence at these punctuation-defined boundaries.

Robust insertion policy. Natural text is irregular: punctuation can be dense (e.g., lists, numbers, ellipses) or absent for long stretches (e.g., code or tables). To keep segment lengths well behaved, we bound the gap between consecutive insertions with two thresholds. A *minimum gap* $g_{\min}$=16 skips



Figure 1: **An overview of Punctuation-Guided Attention (PGA).** The figure illustrates the evolution from a standard autoregressive model to our proposed architecture. *(Top)* A standard Transformer processes a flat sequence of tokens. *(Middle)* Our PAST method (Section 3.1) introduces specialized "summary tokens" at semantic boundaries defined by punctuation ('.' in this instance). *(Bottom)* The final PGA attention pattern, shown as a matrix where colored cells indicate permitted attention, combines local attention (green) with global attention (cyan) restricted to the aggregation points.

a punctuation mark if fewer than $g_{\min}$ tokens have elapsed since the last insertion, preventing dense punctuation from producing redundant summary tokens. A *maximum gap* $g_{\max}$=128 forces an insertion if $g_{\max}$ tokens elapse without an eligible punctuation mark, ensuring that long unpunctuated spans are still summarized. Every segment therefore spans between $g_{\min}$ and $g_{\max}$ tokens.

Crucially, we apply a loss mask to these tokens, ensuring they do not contribute to the language modeling loss during training. This design gives the model the flexibility to learn how best to use these tokens for storing semantic information, rather than forcing them to predict a specific output. Investigating whether specialized auxiliary losses could further guide the representations within these summary tokens is a promising direction for future work (Ghatkesar et al., 2025).

By creating these explicit summary tokens within the sequence, we provide the model with a structured mechanism for performing complex, multi-token reasoning. Instead of reasoning over a flat sequence of individual tokens, the model can operate over a hierarchy of concepts, using the summary tokens to access and compare aggregated information from different parts of the text. As our ablations in Section 5.1 show, the inclusion of summary tokens unlocks new capabilities and measurably improves performance, without increasing the parameter count of the model beyond the embeddings of the summary tokens.

3.2 PGA: Hierarchical Attention for Multi-Token Reasoning

While the summary tokens introduced above provide dedicated space for semantic aggregation, a standard full self-attention mechanism would still require the model to expend effort to differentiate these special tokens from standard tokens. This can add unnecessary complexity and noise to the learning process. To address this and explicitly guide the model's information flow, we introduce a structured and efficient hierarchical attention pattern designed to fully leverage the summary tokens.

Our approach combines local sliding-window attention with a global attention pattern focused on the summary tokens, as visualized in Figure 1 (bottom). Specifically, any given token can attend to two sets of other tokens:

1. **Local context:** It can fully attend to all tokens within a fixed-size local window of size w. This preserves the rich, fine-grained interactions necessary for understanding the immediate local context.
2. **Global context:** For the sequence beyond this local window, its attention is restricted exclusively to the summary tokens of all preceding segments.

This creates an efficient attention pattern where detailed local information is processed locally, while long-range dependencies are captured by attending to the compact, semantic summaries provided by the summary tokens.

This hierarchical structure has two desirable properties. First, it facilitates robust multi-token reasoning across the entire context, as any token can efficiently access the summarized information of all prior segments. Second, it allows each token to more easily tap into the model's high-level understanding of the preceding context without being overwhelmed by noise from every past token. As our ablations in Section 5.1 confirm, this guided attention pattern is a critical component, outperforming models that use summary tokens with standard full attention.

3.3 Efficient Inference via KV Cache Reduction

Beyond improving model performance, our hierarchical architecture directly addresses one of the most significant bottlenecks in deploying large language models: the memory footprint of the KV cache during autoregressive inference. In a standard Transformer, the size of this cache grows linearly with sequence length, consuming large amounts of high-bandwidth memory as context lengths have grown to 128K–1M tokens and beyond.

PGA offers a natural solution to this problem. During inference, it is no longer necessary to retain the keys and values for every token in the context history. Instead, the KV cache need only store the states for two distinct groups of tokens:

1. The tokens within the currently active local sliding window.
2. The summary tokens from all completed semantic segments in the history.

All standard tokens from past segments can be discarded once they fall outside the local window, since later tokens reach them only through the corresponding summary tokens, leading to a substantial reduction in required memory.

To quantify these savings, consider a context of T tokens. Because every segment spans between $g_{\min}$ and $g_{\max}$ tokens (Section 3.1), the number of cached summary tokens K is bounded:

$$\frac{N\,T}{g_{\max}} \;\le\; K \;\le\; \frac{N\,T}{g_{\min}}, \qquad \text{so} \qquad \text{KV cache} \;\approx\; w + K \in \left[\, w + \tfrac{T}{32},\; w + \tfrac{T}{4} \,\right] \tag{1}$$

for $N{=}4$, $g_{\min}{=}16$, and $g_{\max}{=}128$. Relative to the T entries of full attention, the cache is therefore reduced by a factor approaching $4\times$ in the worst case and up to $32\times$ in the best case once $T \gg w$. The cache grows within a fixed, predictable band rather than with the full context, and neither pathological case can break the bound: dense punctuation cannot inflate K beyond $T/4$, and punctuation-free spans cannot starve the model of summaries. In practice, punctuation accounts for about 10% of tokens across ClimbMix (Diao et al., 2025) domains (distribution in Appendix A), an average spacing below $g_{\min}$; the minimum-gap guard is thus frequently active, placing typical text near the dense end of this band, i.e., an approximately $4\times$ reduction.

We also expect computing these summary tokens to introduce minimal latency, for two reasons. First, the summary tokens for a segment are processed in parallel and do not require extra sequential forward passes in the autoregressive loop. Second, and more critically, large-scale LLM inference is often memory-bound rather than compute-bound. The reduction in memory bandwidth required to read the KV cache at each generation step can therefore be expected to offset the minor additional computation. We leave a measured end-to-end latency evaluation in an optimized serving system to future work.

4 Experimental Setup

This section details the experimental framework used to evaluate our method. We first describe a synthetic task created specifically to probe and stress-test the individual contributions of our PAST and PGA components in a controlled setting. We then specify the model architectures and training recipes used for our models and all baselines, and the real-world dataset used to evaluate generalization and convergence.

4.1 SYNTHETIC TASK: MULTI-TOKEN REASONING

To evaluate multi-token reasoning and to stress-test our proposal against relevant prior work, we build upon the synthetic task introduced in Multi-Token Attention (Golovneva et al., 2025). This task is explicitly designed to probe a model's ability to identify a set of non-contiguous items within a long and noisy context.

The task is structured as a sequence-retrieval problem. The input consists of C distinct blocks of context. Each block is a random sequence of S characters, and blocks are separated from each other by a punctuation mark. Following the C context blocks, a final query block is provided, containing L unique characters (where $L < S$). The model's objective is to identify and output the one context block that contains all of the characters in the query. For example, given the input `hjnvt.qfjgt.whftb.bjtpq. ... .pxjvf.ulhik.qoiax#pb`, the fourth block (`bjtpq`) is the response because it contains both `p` and `b`. For this task, we treat '.' and '#' as punctuation.

While conceptually simple, this setup is challenging because it requires the model to perform a conditional check across multiple, distributed tokens, rather than matching a single token or a contiguous phrase. To study scalability and robustness, we create multiple versions of the task by varying the number of context blocks (C), the number of characters within each block (S), and the query length (L). This controlled environment allows us to analyze how PAST (aggregation) and PGA (attention) enable the model to succeed as the context becomes longer and more complex (Section 5.1).

4.2 MODEL ARCHITECTURE

On the synthetic reasoning task, we adopt the Transformer architecture (Vaswani et al., 2017) used in Multi-Token Attention (Golovneva et al., 2025) as our baseline model. To create our variants, we extend this architecture with the PAST mechanism by adding three new, unique summary tokens (N=3) to the model's vocabulary; at each punctuation mark, these summary tokens are inserted into the sequence. Our ablation studies found that using distinct embeddings for these summary tokens consistently outperformed using a single, repeated summary token, so we adopt the former strategy (Section 5.1).

For our experiments on real-world data, we build upon the Llama 3.2 1B architecture (Grattafiori et al., 2024). We test two model scales: (1) the original 1B-parameter model (training curves in Appendix B), and (2) a smaller 256M-parameter variant, constructed by reducing the hidden dimension to 1024 while keeping other hyperparameters, such as the number of layers, consistent with the base model. We use the ClimbMix dataset (Diao et al., 2025) with the cluster weights recommended in that work.

4.3 TRAINING RECIPE

For the synthetic reasoning task, all models are trained from scratch for 100,000 steps with a global batch size of 64 sequences, using the AdamW optimizer (Loshchilov & Hutter, 2019) with a constant learning rate of 10^{-4}.

For our experiments on real-world data, we evaluate both continued pretraining (primary evaluation) and pretraining from scratch on a corpus of approximately 10 billion tokens, to analyze downstream performance and convergence characteristics, respectively. All models, including baselines, are trained with the standard cross-entropy objective. These large-scale runs use a global batch size of 4 million tokens and a standard cosine learning-rate schedule with linear warm-up. For this study, we treat as punctuation boundaries all tokens that contain one of the following characters: `.` `,` `!` `?` `:` `;` and newline, resulting in 8,889 unique token candidates, with summary tokens inserted according to the policy in Section 3.1.

Table 1: **Impact of PAST and PGA on multi-token reasoning.** Accuracy (%) on the synthetic task (Section 4.1) across block sizes (S), block counts (C), and query lengths (L). PAST Unique uses distinct embeddings for each summary token.

Model	S=5, C=40 L=2	S=7, C=48 L=5	S=10, C=40 L=5	S=5, C=160 L=2
Transformer	≈0.3	≈0.2	≈0.2	≈0.2
PAST	99.38	≈0.3	≈0.3	≈0.3
PAST Unique	99.38	98.68	≈0.3	≈0.3
PGA	**99.96**	**99.59**	**99.45**	**96.50**

Table 2: **Continued pretraining benchmark performance.** Llama 3.2 1B continued-pretrained with and without PGA on identical data. PGA improves accuracy on all seven tasks within a small number of training steps.

Model	ARC-Challenge	HellaSwag	LogiQA	MMLU	PubMedQA	SciQ	WinoGrande
Baseline	38.2	64.15	27.2	35.8	65.0	91.3	58.6
PGA	**41.0**	**65.4**	**29.2**	**38.0**	**67.8**	**91.8**	**62.1**

5 RESULTS

We first evaluate the core multi-token reasoning capabilities of PAST and PGA on the targeted synthetic task (Section 5.1). We then evaluate the full approach on real-world data in two settings: continued pretraining (Section 5.2) and pretraining from scratch (Section 5.3).

5.1 SYNTHETIC TASK: MULTI-TOKEN REASONING

Table 1 summarizes results on the synthetic multi-token reasoning task. First, consistent with prior work (Golovneva et al., 2025), the standard self-attention baseline struggles on this task despite theoretically having full visibility of the history. This validates our intuition that an all-to-all attention mechanism finds it exceedingly hard to draw conclusions that depend on multiple, disparate tokens, especially as context length and task complexity grow.

Introducing summary tokens via PAST provides a substantial improvement, demonstrating that dedicated aggregation points help the model perform multi-token deductions. However, this model's performance degrades as task complexity increases (S=7, C=48 and beyond). To investigate further, we increased the capacity of the summary tokens by giving each a distinct embedding (PAST Unique). This allowed the model to scale to the S=7, C=48 setting (finding patterns across 48 blocks of 7 tokens each), but not to longer blocks (S=10) or many more blocks (C=160).

We hypothesize that this is due to the noise inherent in standard self-attention: even with summary tokens, the model struggles to identify and learn from the most relevant information. Our PGA attention pattern mitigates this. By combining PAST with punctuation-guided attention, the model exhibits strong scalability and high accuracy across all tested configurations, handling both longer contexts and reasoning across more tokens.

5.2 DOWNSTREAM BENCHMARKS: CONTINUED PRETRAINING

To demonstrate the generality of our method, we apply it to an existing Llama 3.2 1B model (Grattafiori et al., 2024) via continued pretraining. As shown in Table 2, our approach yields consistent gains over a baseline trained on identical data across all seven benchmarks, confirming the broad applicability of the architectural change. Notably, these improvements are achieved in just 4.5K training steps, a small fraction of the compute used for the model's original pretraining.

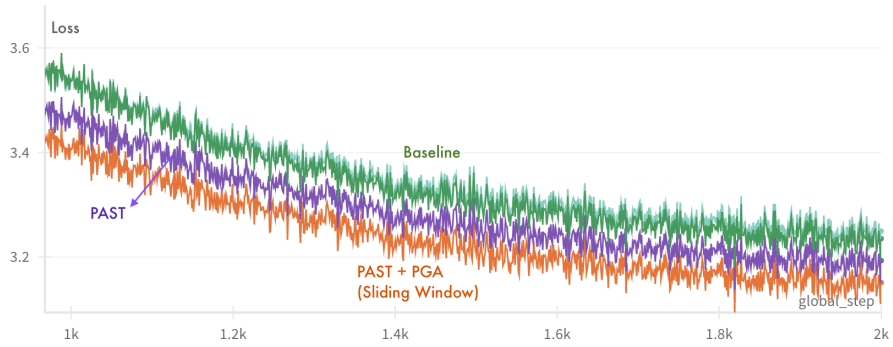


Figure 2: **Training loss on real-world data (256M parameters).** Training loss versus training steps for three variants: the baseline Transformer (green); PAST, which inserts summary tokens but uses standard full attention (purple); and PAST + PGA, which adds punctuation-guided sliding-window attention (orange). Introducing PAST summary tokens alone leads to lower training loss than the baseline, even though these tokens receive no direct loss signal. The full model achieves the best convergence.

Table 3: **Pretraining-from-scratch benchmark performance (256M parameters).** Comparison of PGA against a baseline, both trained from scratch on identical data. PGA improves accuracy on six of seven tasks.

Model	ARC-Challenge	HellaSwag	LogiQA	MMLU	PubMedQA	SciQ	WinoGrande
Baseline	24.1	29.5	**26.9**	25.3	42.8	63.6	49.5
PGA	**26.2**	**33.3**	25.65	**25.5**	**50.0**	**77.4**	**51.1**

5.3 Convergence on Real-World Data: Pretraining from Scratch

To assess our approach on real-world data, we present an analysis of a 256M-parameter model pretrained on approximately 10 billion tokens from ClimbMix (Diao et al., 2025). We focus on training loss convergence (Figure 2) and downstream performance (Table 3).

First, introducing the PAST mechanism alone consistently improves training convergence over the baseline throughout the entire run, suggesting that dedicated summary tokens for semantic aggregation are inherently beneficial, even without a specialized attention pattern or auxiliary losses to explicitly leverage them. More compellingly, the full PAST + PGA model yields the best training convergence and the lowest overall loss. This provides initial evidence for our central hypothesis: the gains come from a semantically rich KV cache (PAST) combined with minimal distraction from filler tokens (PGA). The same trend holds at 1B parameters (Appendix B). Downstream, PGA improves accuracy on six of seven benchmarks, with the largest gains on SciQ and PubMedQA.

6 Conclusion

In this work, we introduced Punctuation-Guided Attention (PGA) to address a critical limitation of the Transformer architecture: efficiently reasoning over multiple pieces of context. PGA leverages specialized summary tokens at natural semantic breaks (PAST) together with a hierarchical attention pattern for structured, long-range reasoning. Our experiments show strong performance on a synthetic multi-token reasoning task where standard attention fails, as well as improvements across a range of real-world benchmarks in both continued pretraining and pretraining from scratch, while enabling a substantially smaller KV cache at inference. Our work demonstrates that by aligning architectural design with the semantic structure of language, we can build more capable and efficient models.

AI USE STATEMENT

In this work, we have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, designing research methodology or experiments, implementing methods, cleaning or reformatting datasets, or interpreting results; translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to reformat the manuscript into the conference template, reconstruct the bibliography and format references, edit the paper to improve readability, and suggest the title and revisions to the abstract. All AI-assisted text, references, and formatting were reviewed and verified by the authors against the original manuscript and cited sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The synthetic task construction is described in Section 4.1. The summary-token insertion policy and attention mask are specified in Sections 3.1 and 3.2. Model architectures, the training data mixture, training hyperparameters, and the punctuation set used for segmentation are given in Sections 4.2–4.3, and the punctuation frequency distribution underlying our KV cache analysis is shown in Appendix A. All real-data experiments use the publicly available ClimbMix dataset and the Llama 3.2 architecture.

REFERENCES

Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, Mostofa Patwary, Yingyan Lin, Jan Kautz, and Pavlo Molchanov. CLIMB: Clustering-based iterative data mixture bootstrapping for language model pre-training, 2025. URL https://arxiv.org/abs/2504.13161.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, and Mao Yang. SeerAttention: Learning intrinsic sparse attention in your llms, 2025. URL https://arxiv.org/abs/2410.13276.

Aarti Ghatkesar, Uddeshya Upadhyay, and Ganesh Venkatesh. Looking beyond language priors: Enhancing visual comprehension and attention in multimodal models, 2025. URL https://arxiv.org/abs/2505.05626.

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. Multi-token attention, 2025. URL https://arxiv.org/abs/2504.00927.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens, 2024. URL https://arxiv.org/abs/2310.02226.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition, 2020. URL https://arxiv.org/abs/2005.08100.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention, 2024. URL https://arxiv.org/abs/2407.02490.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack, 2024. URL https://arxiv.org/abs/2406.10149.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A context-aware sparse attention mechanism for efficient long-sequence inference, 2025. URL https://arxiv.org/abs/2502.20766.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens, 2024. URL https://arxiv.org/abs/2304.08467.

Anton Razzhigaev, Matvey Mikhalchuk, Temurbek Rahmatullaev, Elizaveta Goncharova, Polina Druzhinina, Ivan Oseledets, and Andrey Kuznetsov. LLM-Microscope: Uncovering the hidden role of punctuation in context memory of transformers, 2025. URL https://arxiv.org/abs/2502.15007.

Lin Song, Yukang Chen, Shuai Yang, Xiaohan Ding, Yixiao Ge, Ying-Cong Chen, and Ying Shan. Low-rank approximation for sparse attention in multi-modal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13763–13773, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions, 2019. URL https://arxiv.org/abs/1901.10430.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon, 2024. URL https://arxiv.org/abs/2401.03462.

Qianchao Zhu, Jiangfei Duan, Chang Chen, Siran Liu, Xiuhong Li, Guanyu Feng, Xin Lv, Huanqi Cao, Xiao Chuanfu, Xingcheng Zhang, Dahua Lin, and Chao Yang. SampleAttention: Near-lossless acceleration of long context llm inference with adaptive structured sparse attention, 2024. URL https://arxiv.org/abs/2406.15486.

A Punctuation Frequency Distribution

B Training Loss at 1B Parameters

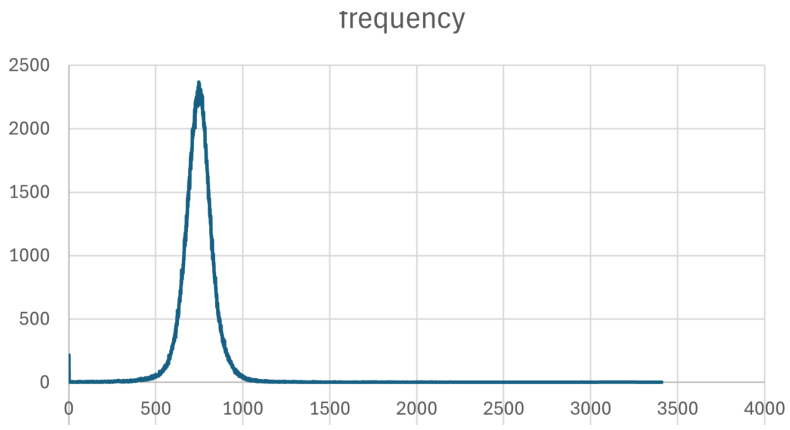


Figure 3: **Punctuation frequency distribution in ClimbMix.** Number of punctuation boundaries observed across ClimbMix samples, counting `.` `,` `!` `?` `:` `;` and newline.

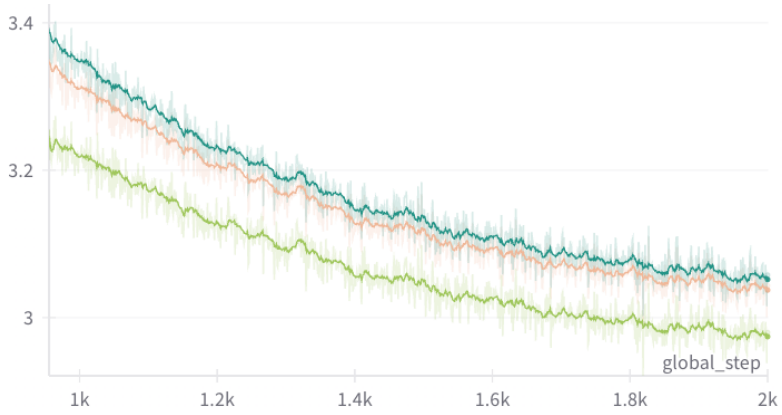


Figure 4: **Training loss on real-world data (1B parameters).** Trends are consistent with those observed for the 256M-parameter model (Figure 2): combining PAST with the PGA attention pattern achieves the best convergence.