HEAD DIMENSION MATTERS IN LANGUAGE MODELS FOR LONG-CONTEXT RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-head attention retrieves and combines context representations through query–key interactions and value aggregation, with head dimension constraining the per-head representation space. In long-context processing, it remains unclear whether allocating a fixed parameter budget to fewer, wider heads improves information retrieval. In this paper, we investigate how head dimension affects the performance of long-context retrieval tasks. Our empirical studies reveal that pretraining loss and short-context non-retrieval task performance are not affected much by head dimension, while long-context retrieval performance improves with wider heads at fixed model sizes. We find that models with wider heads have a higher proportion of retrieval heads, associating more parameters with retrieval. Our ablation studies also show that the retrieval performance improves with wider heads even when the projection weights in attention modules are frozen during long-context continual pretraining, suggesting that wider heads not only benefit updates in attention modules but lead to more effective adaptation of components such as embeddings and MLPs.

1 INTRODUCTION

Large language models (LLMs) are increasingly expected to process long texts including documents, multi-turn conversations, and code repositories, as well as longer reasoning traces used for test-time scaling (Muennighoff et al., 2025). Although modern LLM architectures support processing long inputs, a large supported context window does not necessarily imply effective use of information throughout the context (Hsieh et al., 2024; Liu et al., 2024). It is therefore increasingly important to explore improvements in architectures and training recipes for enhanced long-context capabilities.

The ability to retrieve task-relevant information from a large amount of context is a fundamental prerequisite for effective long-context processing (Hsieh et al., 2024; Yen et al., 2025; Qiu et al., 2025). Recent work suggests that successful retrieval is not uniformly implemented across a Transformer: a small subset of attention heads has been shown to play a central role in extracting relevant information from long inputs (Wu et al., 2025; Zhang et al., 2025).

These findings motivate a specific architectural question: *under a fixed attention parameter budget, how does allocating capacity to fewer, wider heads affect long-context retrieval?* In this work, we study the relationship between long-context retrieval capability and the head dimension of self attention modules. We train 0.6B and 3B parameter models from scratch through a simplified training pipeline consisting of pretraining (PT) and long-context continual pretraining (CPT) with web and code-centric data mixes, and additionally apply supervised fine-tuning (SFT) to 3B models to examine whether head dimension affects performance on short-context, non-retrieval tasks. Parameter-matched comparisons of models with identical architectures except for the balance between the number of heads and the head dimension reveal that wider heads lead to better performance on retrieval tasks.

To investigate the performance gap between models with different head dimensions, we first measured the retrieval scores of attention heads following Wu et al. (2025). We found that wider-head models have a higher proportion of retrieval heads, with comparable absolute retrieval-head counts despite fewer heads overall. This indicates that wider-head models associate more parameters with retrieval heads. Comparisons between pretrained checkpoints and their corresponding CPT checkpoints further suggest that much of this association is already established during pretraining.

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.

We then repeated CPT with frozen projection weights in attention modules to examine how head dimension affects long-context adaptation. The improvement in retrieval performance after CPT is larger in models with wider heads even with frozen projections in attention, whereas updating all the projections provides similar additional improvements across head dimensions. Further analyses show that the sharpness of attention and changes in token-level attention rankings do not fully explain the performance gap. These results suggest that wider heads improve long-context retrieval by enabling more effective adaptation of the remaining trainable components, including embeddings and MLPs, rather than through greater plasticity of the attention modules themselves.

2 EXPERIMENTAL SETUP

We trained decoder-only Transformers from scratch through a three-stage pipeline consisting of PT, CPT, and SFT. We use models with 0.6B parameters for ablation experiments, and use models with 3B parameters to verify that the comparison is also observed at a larger model scale. Both types of models consist of sliding window attention (SWA) layers with a window size of 512 tokens and full attention layers that attend to all past key-values, interleaved in 3:1 ratio. We apply standard RoPE to SWA heads and 64 dimensional partial RoPE to full attention heads. For partial RoPE, we use a small base $\theta = 100$, with which we can use the same θ without additional scaling during context extension.

For an input sequence $X \in \mathbb{R}^{L\times D}$ with sequence length L and model dimension D, multi-head attention (MHA) with H heads is written as

$$\mathrm{MHA}(X) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_H)W_O,$$
$$\mathrm{head}_h = \mathrm{softmax}\left(\frac{XW_Q^{(h)\top}W_K^{(h)}X^\top}{\sqrt{d}} + M\right)XW_V^{(h)\top},$$

where d is the head dimension, $W_Q^{(h)}, W_K^{(h)}, W_V^{(h)} \in \mathbb{R}^{d\times D}$, $W_O \in \mathbb{R}^{Hd\times D}$, and M represents the causal or sliding window mask. The number of heads H and head dimension d are standard architectural hyperparameters in MHA. For a fixed model dimension D, the number of attention parameters scales as $\mathcal{O}(DHd)$ and dense attention computation scales as $\mathcal{O}(L^2Hd)$. Our models use grouped-query attention, where $W_K^{(h)}$ and $W_V^{(h)}$ are shared among heads in the same group. We use a fixed query-to-KV head ratio, so the KV-cache size also scales with Hd. Thus, when Hd is fixed, changing the number and size of heads leaves the attention parameter count, computation, and KV-cache size essentially unchanged, while changing how the same parameter and compute budgets are allocated across heads.

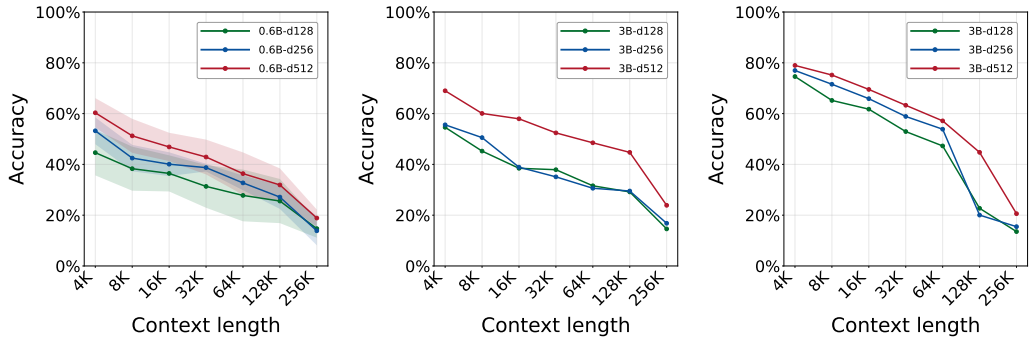
For each model scale, we prepare three architectures that are identical except for the number of heads and the dimension of each head in full attention layers as shown in Table 1. For the symmetric configurations at each model scale, the parameter allocation and key-value cache size are identical. For the 0.6B model family, we conduct additional experiments on models with asymmetric head size with distinct QK and V dimensions, whose parameter count is slightly larger.

Table 1: Architecture names and their configurations. Dimensions and head counts refer to full-attention layers. The query and key dimensions are always identical, whereas the value dimension may differ. Full configurations are provided in Appendix A.

Model Name	Params	Q/K Dim.	V Dim.	#Q Heads	#KV Heads
`0.6B-d128`	0.6B	128	128	16	4
`0.6B-d256`	0.6B	256	256	8	2
`0.6B-d512`	0.6B	512	512	4	1
`0.6B-k512-v128`	0.6B	512	128	8	2
`0.6B-k128-v512`	0.6B	128	512	8	2
`3B-d128`	3B	128	128	32	8
`3B-d256`	3B	256	256	16	4
`3B-d512`	3B	512	512	8	2

Table 2: RULER-100 overall scores averaged over 13 tasks and seven lengths from 4K to 256K. Scores are on a 0–100 scale. The 0.6B results are means ± sample standard deviations across three runs; each 3B allocation has one run.

Model size	Stage	d128	d256	d512	k512-v128	k128-v512
0.6B	CPT	31.25 ± 7.75	35.46 ± 3.60	**41.21** ± 5.86	37.78 ± 4.15	31.28 ± 2.61
3B	CPT	35.93	36.72	**50.93**	-	-
3B	SFT	48.28	51.81	**58.49**	-	-



(a) 0.6B (CPT) (b) 3B (CPT) (c) 3B (SFT)

Figure 1: RULER-100 scores by context length. Each point averages 13 tasks at the indicated context length on a 0–100 scale. In (a), lines and bands show the mean and ± sample standard deviation across three 0.6B runs. In all the settings, d512 achieves the highest (average) score at every evaluated length.

In the PT and CPT stages, we use data mixes consisting of web (60%), code (30%), and SFT-like (10%) datasets. The sequence length is set to 4,096 in PT and 262,144 in CPT. We use 100B tokens for PT and 8B tokens for CPT. We apply SFT to the 3B models using short and long SFT trajectories with the same token ratio, totaling 1B tokens.

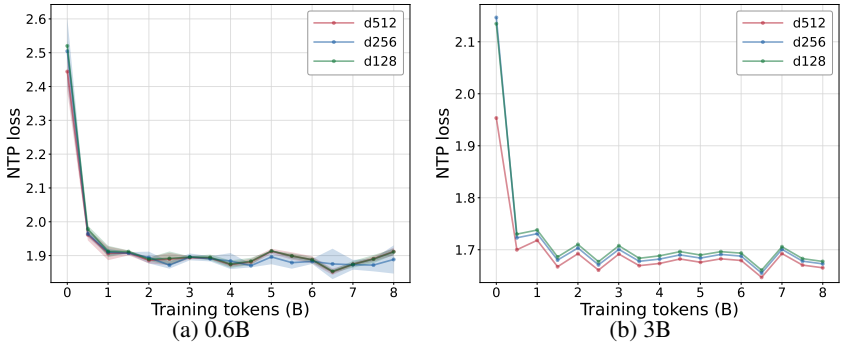
We use the AdamW optimizer (Loshchilov & Hutter, 2019) for all the training stages. In the PT and SFT stages, the learning rate is warmed up from zero to the maximum learning rate and held constant for the remaining steps. In the CPT stage, we skip the learning rate warm-up and use a constant learning rate. Additional details of architectures, data mixes, and training setup are provided in Appendix A.

Evaluation: We measure the long-context retrieval capability with the RULER benchmark (Hsieh et al., 2024). We evaluate 13 RULER tasks, comprising eight needle-retrieval variants, variable tracking, common-word extraction, frequent-word extraction, and two question-answering tasks. We evaluate all of these tasks with context lengths ranging from 4,096 to 262,144. We only generate 100 examples for each task to keep the evaluation cost low. We call this variant *RULER-100*, which we confirmed does not affect the stability of the evaluation.

3 Model Performance Comparisons

We compare the head allocations in Table 1 to assess whether wider heads improve long-context retrieval after CPT. We examine RULER performance alongside next-token prediction (NTP) training loss to determine whether the two metrics favor the same head allocation.

Wider heads improve retrieval at a fixed attention budget. Table 2 and Figure 1 show the average and length-wise RULER-100 scores, respectively. The d512 models achieve the highest average scores in all the settings and context lengths. On the average scores at 0.6B parameter scale, d512 exceeds d128 and d256 by 9.96 and 5.75 points, respectively; at 3B parameter scale, the gains are 15.00 and 14.21 points.



(a) 0.6B (b) 3B

Figure 2: training NTP loss during 8B-token CPT at 256K. The horizontal axis counts CPT tokens; the vertical axis is training cross-entropy. In (a), lines and bands are the mean and ± 1 sample standard deviation across three 0.6B runs. Lower training loss does not always correspond to higher RULER performance.

Table 3: Short-context task performance for the 3B models after SFT. Abbreviations: ARC-C (ARC-Challenge), ARC-E (ARC-Easy), HS (HellaSwag), WG (WinoGrande), OBQA (OpenBookQA), CSQA (CommonsenseQA), Avg (Average).

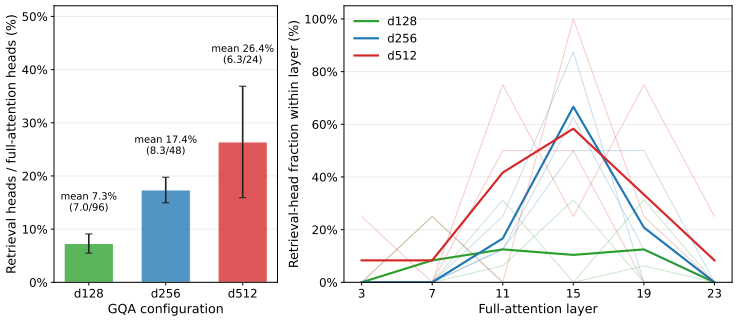
Head Size	ARC-C	ARC-E	HS	WG	RACE	PIQA	OBQA	CSQA	Avg
`d128`	35.6	62.3	58.2	57.4	36.2	73.0	36.6	38.1	49.7
`d256`	**35.8**	61.0	58.1	**59.4**	35.0	73.3	37.4	28.9	48.6
`d512`	33.9	**64.0**	**58.7**	57.9	**37.4**	**73.7**	**39.0**	**39.6**	**50.5**

These results favor allocating the fixed attention parameter budget to fewer, wider heads for long-context retrieval. The advantage is not uniform across individual tasks, and all allocations still degrade at the longest lengths; Appendix B.2 provides the breakdown.

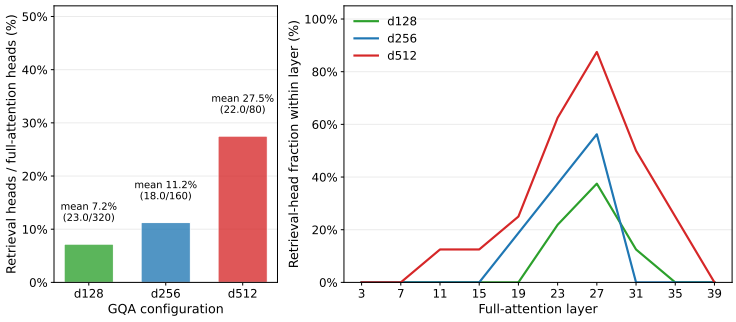
Training loss does not fully capture the head-dimension effect. Figure 2 shows the CPT training NTP loss. PT losses also vary little across head dimensions, as shown in Table 8. At 0.6B parameter scale, `d256` has the lowest mean final loss, yet `d512` has the highest overall RULER-100 score. At 3B parameter scale, `d512` is best on both metrics. Thus, training loss alone does not consistently identify the head allocation with the best long-context retrieval performance. This is consistent with prior findings that standard token-averaged perplexity is not a reliable proxy for long-context downstream performance (Hu et al., 2024; Fang et al., 2025). Final losses and PT trajectories are reported in Appendix B.1.

Larger QK dimensions benefit retrieval more than larger V dimensions. Among the 0.6B models, `d512` achieves the highest mean RULER-100 score, followed by `k512-v128` and `d256`, while `d128` and `k128-v512` have nearly identical mean scores (Table 2). The two asymmetric configurations have the same head counts, and their performance gap favors larger QK over larger V dimensions for long-context retrieval. At a QK dimension of 512, the higher mean score of `d512` relative to `k512-v128` suggests an additional benefit from a larger V dimension. At a QK dimension of 128, however, `k128-v512` shows essentially no gain over `d128`. Although the latter two comparisons also change head count, this pattern is consistent with a QK bottleneck: narrow QK representations may constrain the selection of relevant positions, limiting the benefit of additional V capacity for conveying information from those positions.

The retrieval performance is not traded-off with short-context task performance. To assess whether a trade off exists between long-context retrieval capabilities and short-context task performance, we evaluate 3B models after SFT with short-context downstream tasks. The task set consists of question answering (ARC-Easy and ARC-Challenge (Clark et al., 2018)), general commonsense reasoning (CommonsenseQA (Talmor et al., 2019), WinoGrande (Sakaguchi et al., 2019)), commonsense NLI (Hellaswag (Zellers et al., 2019)), physical commonsense reasoning (PIQA (Bisk et al., 2020)), reading comprehension (RACE (Lai et al., 2017)), and scientific QA (OpenBookQA



(a) 0.6B



(b) 3B (after SFT)

Figure 3: Retrieval-head fractions in the (a) 0.6B and (b) 3B models. Left: the fraction of full-attention query heads classified as retrieval heads; annotations give the retrieval-head count relative to the total number of eligible heads. Right: retrieval-head fractions within individual full-attention layers, using zero-based layer indices. The 0.6B results are means across three runs; faint lines show individual runs. Each 3B model has one run. Wider-head models exhibit higher retrieval-head fractions at both scales, while absolute retrieval-head counts remain comparable across allocations.

(Mihaylov et al., 2018)). The results are shown in Table 3. The model with a head dimension of 512 performs the best on average, while we observe no clear monotonic trend with the head dimensions. We can at least conclude that head dimension does not affect the short-context tasks negatively, and the improvement in long-context retrieval capability is not traded-off with the short-context performance.

4 Analysis of Retrieval Heads

To assess the mechanistic differences caused by changing head dimension, we measured the number of retrieval heads in each full attention layer. We use the single-needle variants of needle-in-a-haystack tasks in RULER and measured the retrieval score of each full-attention head. Following Wu et al. (2025), we count a head with the retrieval score ≥ 0.1 as a retrieval head. The results in Figure 3 reveal that models with wider heads allocate a higher ratio of full-attention heads for retrieval. Note that the number of retrieval heads does not change significantly despite of fewer total number of heads in models with wider heads. This indicates that the models with wider heads associate more parameters with retrieval heads in total, which may lead to higher accuracy in attending to needles due to the larger capacity of retrieving queries.

4.1 Retrieval-head emergence before and after CPT

To examine how retrieval heads change through CPT, we compare the 100B-token PT checkpoints with their corresponding 8B-token full-parameter CPT checkpoints for the three symmetric 0.6B-parameter configurations. The PT checkpoints are evaluated with the inference-time attention scaling described in Appendix A.4, without updating any model parameters, whereas CPT checkpoints use no additional scaling. We identify retrieval heads using the same single-needle RULER protocol

Table 4: Retrieval-head counts before and after CPT for the 0.6B models. PT denotes the 100B-token PT checkpoint evaluated with inference-time attention scaling (Appendix A.4); CPT denotes the corresponding checkpoint after 8B tokens of full-parameter CPT, evaluated without additional scaling. The rightmost column reports the number of retrieval heads common to both stages, matched by layer and query-head index within each run. All the counts are computed within each run and then averaged across three runs per configuration.

Model	PT (attention scaling)	CPT	Shared
`0.6B-d128`	4.00	7.00	4.00
`0.6B-d256`	8.33	8.33	8.00
`0.6B-d512`	5.00	6.33	5.00

and retrieval-score threshold (≥ 0.1) as above. Table 4 reports the numbers of detected and shared full-attention query heads averaged over three runs.

For `d128`, the mean retrieval-head count increases from 4.00 in scaled PT to 7.00 after CPT, with all the retrieval heads detected in PT checkpoints retained. The retrieval-head fraction accordingly rises from 4.2% to 7.3%, indicating that the capacity allocated to detected retrieval heads is even smaller at PT than after CPT.

The changes in detected counts are smaller for the wider-head models. For `d256`, the mean count is 8.33 at both stages, with 8.00 shared heads. For `d512`, the mean count increases modestly from 5.00 to 6.33, with all the retrieval heads detected in PT checkpoints retained. Thus, much of the retrieval heads in these models are already present at the end of pretraining. Run-level counts and shared membership are reported in Appendix C.

5 Head dimension and long-context adaptation

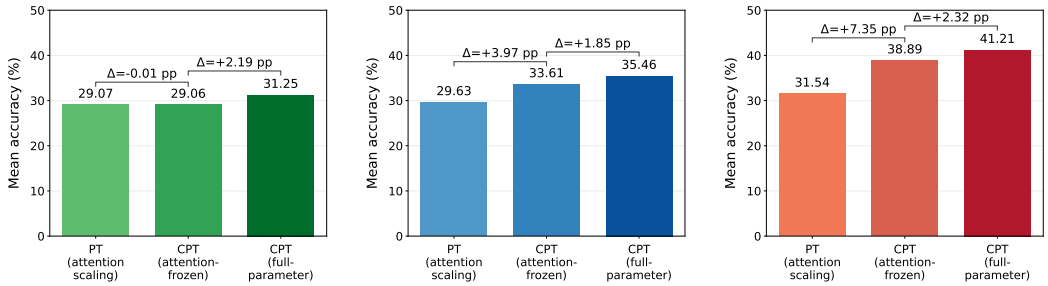
The analyses provided above characterize how head dimension affects retrieval capabilities; here, we examine how models with different head dimensions adapt during CPT. Because updates on query, key, value, and output projections can directly alter the behavior of attention modules, we first isolate their contribution by freezing these projections during CPT in Section 5.1. We then examine two complementary properties of the resulting attention maps in Section 5.2: the concentration of attention probabilities and the ranking of attended token positions. Together, these analyses test whether adaptation within attention modules accounts for the differences in retrieval performance across head dimensions.

5.1 Retrieval performance gains of CPT with frozen attention weight matrices

For each symmetric 0.6B parameter configuration—`d128`, `d256`, and `d512`—we conduct *full-parameter CPT* (Full CPT) and *attention-frozen CPT* (Frozen CPT) from PT checkpoints of three independent runs. Within each run, both conditions start from the same 100B-token PT checkpoint and undergo 8B tokens of CPT at a sequence length of 256K. In Frozen CPT, we fix W_Q, W_K, W_V, W_O in both full-attention and sliding-window attention layers and only update other parameters; see Table 9 for the detailed parameter-update policies. Note that fixed projections do not imply fixed attention maps: their inputs and normalization gains can still adapt.

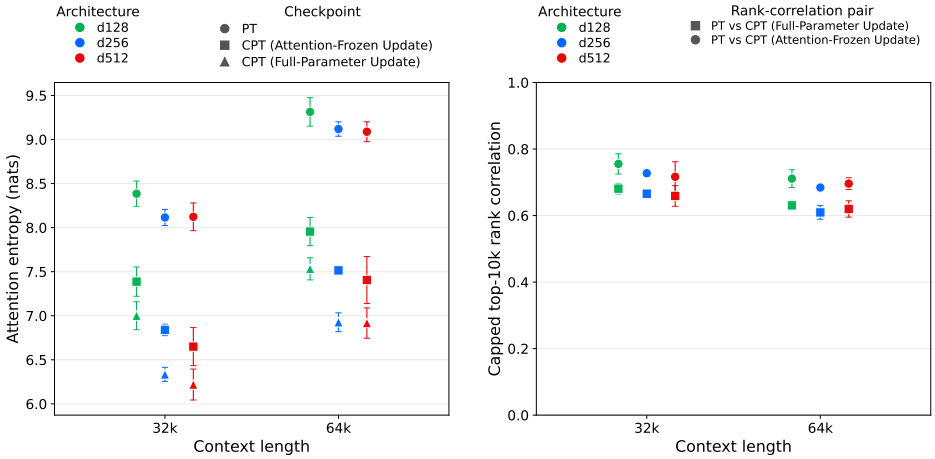
We evaluate RULER-100, averaging over 13 tasks and seven context lengths from 4K to 256K. As a performance baseline, we evaluate PT checkpoints with inference-time attention logit scaling (*scaled PT*; see Appendix A.4), which requires no parameter updates. Neither CPT condition uses this additional scaling. Scores are averaged across three runs. The result for each task and length is provided in Figure 13 and Figures 14–16.

The head-dimension dependence persists without projection updates. Figure 4 shows that the mean improvement of Frozen CPT over scaled PT increases with head dimension: -0.01, $+3.97$, and $+7.35$ points for `d128`, `d256`, and `d512`, respectively. In contrast, the additional mean benefit of Full CPT over Frozen CPT is similar across configurations: 2.19, 1.85, and 2.32 points. The



(a) `0.6B-d128` (b) `0.6B-d256` (c) `0.6B-d512`

Figure 4: Retrieval performance with and without attention weight matrices updates for the three 0.6B-parameter configurations. Bars show means across three training runs, each evaluated and averaged over 13 RULER-100 tasks and seven context lengths from 4K to 256K. The PT baseline uses inference-time attention scaling; neither CPT condition uses additional scaling. Brackets show mean differences in score points, computed before rounding.



(a) attention entropy (b) rank correlation

Figure 5: Attention-map diagnostics on shared RULER examples at context lengths of 32K and 64K, using queries predicting the first three reference-answer tokens. (a) Mean attention entropy; lower values indicate more concentrated attention. (b) Capped top-10k rank correlation between each CPT checkpoint and the corresponding PT checkpoint; higher values indicate more similar token prioritization. PT checkpoints are evaluated without attention scaling. Points and error bars show means and ± 1 sample standard deviation across three training runs, after averaging within each run.

strong head-dimension dependence therefore appears in performance gains achievable while projections in attention modules remain fixed, rather than in the incremental benefit of updating them.

Full CPT also increases the trainable parameter fraction from 83.4% to 100%, without changing model size. Its roughly two-point advantage is compatible with a general benefit from optimizing additional parameters, rather than a benefit that grows with head dimension. The comparison does not isolate trainable parameter count from parameter identity, so it cannot establish that count alone causes the improvement. Nevertheless, these results do not support that the performance gap in retrieval tasks among models with different head dimensions is explained by the benefit of updates in attention weight matrices.

5.2 How does head dimension affect attention maps and model representations?

We next examine whether changes in attention maps explain the performance gains of wider heads under frozen attention weight matrices. For Figure 5, we compare PT without attention scaling

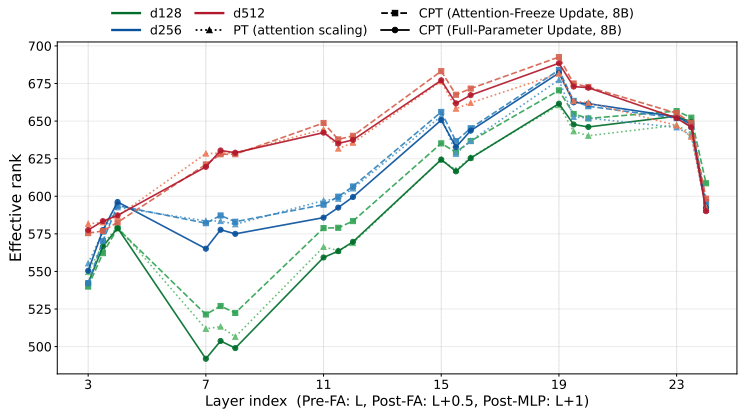


Figure 6: Residual-stream effective rank at 32K context length for the three symmetric 0.6B configurations under scaled PT, Frozen CPT, and Full CPT. Each point is computed from the mean singular-value spectrum over 300 shared examples and three training seeds. For a full-attention layer with zero-based index L, the effective ranks of representations at Pre-FA, Post-FA, and Post-MLP are plotted at the layer index L, $L + 0.5$, and $L + 1$, respectively. These offsets distinguish measurement sites within the same block, not additional layers.

with the corresponding 8B-token Full and Frozen CPT checkpoints on six RULER tasks, using 300 shared examples at each of 32K and 64K. We analyze full-attention queries predicting the first three reference-answer tokens under teacher forcing. We report means and sample standard deviations across three runs after within-run aggregation. See Appendix D.1 for details.

Lower entropy does not fully explain the wider-head advantage. For attention probabilities a_j over causally visible token positions, we measure entropy in nats as $H = -\sum_j a_j \log a_j$. Lower entropy indicates more concentrated attention, but does not establish whether the attended positions are relevant. Figure 5(a) shows lower mean entropy after CPT for every head dimension, with Full CPT below Frozen CPT and Frozen CPT below PT at both context lengths. The higher post-CPT entropy of `d128` is consistent with the lower retrieval performance as it indicates less accurate information retrieval from the context. However, `d256` and `d512` have similar mean entropies under each CPT checkpoint despite their performance gap. Taken together, these results suggest that sharper attention may contribute to retrieval improvements, but average probability concentration alone does not explain the head-dimension effect.

Greater token reordering does not consistently favor wider heads. We compare token prioritization between PT and each CPT checkpoint using *capped top-10k rank correlation*. For each matched layer, query head, and answer query on the same example, we rank all causally visible token positions by decreasing attention probability in each checkpoint. We replace ranks beyond $K = 10{,}000$ with $K + 1$ and compute Pearson correlation between the resulting rank vectors, aligned by token position. Thus, lower-ranked positions remain in the comparison, but differences in their ordering are ignored. Lower correlation indicates greater changes in which tokens receive high attention and how they are ordered, rather than changes in probability magnitudes.

Figure 5(b) shows greater reordering under Full CPT than under Frozen CPT, but differences across head dimensions are small and do not consistently track retrieval performance. At 64K, `d256` exhibits more reordering than `d512` under both CPT conditions despite its lower overall RULER-100 score. Thus, this measure of token reordering does not fully explain the wider-head advantage.

Head dimension shapes the model's internal representations. We examine the effective rank (Roy & Vetterli, 2007) of the residual stream at 32K context length for the three 0.6B configurations, using prompt-token representations before and after attention and after the MLP in each full-attention block. Figure 6 shows that wider-head models generally have higher effective ranks in intermediate blocks, particularly at layers 7–19. This separation is already visible in scaled PT and persists under Frozen CPT, so it is not specific to updates on projection weights in attention modules during CPT. These results associate head allocation with substantial differences in residual-stream

structure, showing that the influence of head dimension extends beyond attention modules to the internal representations propagated through the model. See Appendix D.2 for the details.

Implications. Together, Figures 4 and 5 show that wider heads achieve larger mean retrieval performance gains even with fixed projection weights in attention modules, whereas the measured concentration and reordering of attention maps do not fully account for those gains. Figure 6 further shows that head allocation is associated with differences in residual-stream spectral structure even under Frozen CPT. These results point to an interaction between head allocation and adaptation of the remaining trainable components, including embeddings and MLPs, as a plausible explanation, rather than a larger direct benefit from updating attention weight matrices.

6 Related work

Head dimension and the allocation of attention capacity. Multi-head attention aggregates information through distinct attention distributions and representation subspaces (Vaswani et al., 2017). At fixed total projection width, increasing the number of heads reduces each head's dimension. Bhojanapalli et al. (2020) show that low-dimensional query–key parameterizations can restrict the attainable attention distributions, motivating head dimension as a capacity constraint rather than merely an implementation choice. Conversely, theoretical advantages of multiple heads have been established for memorization and in-context linear regression (Mahdavi et al., 2024; Cui et al., 2024). Head-count requirements also arise in restricted retrieval and multi-task computation settings (Yu et al., 2026; Hsu & Xu, 2026). These results establish task-dependent trade-offs, not a universally optimal head allocation. Our experiments examine this trade-off for long-context language models trained from scratch, keeping total query and key/value projection widths and the per-token KV-cache size fixed. This differs from changing the degree of key/value sharing, as in grouped-query attention (Ainslie et al., 2023). Note that, in most open-weight LLM families published with multiple parameter scales, the number of heads in attention layers is increased in larger models, while the dimension of each head is kept fixed (Grattafiori et al., 2024; Yang et al., 2025; Team, 2026).

Long-context training and architectural adaptation. Prior work improves long-context adaptation through data selection and domain balancing (Fu et al., 2024; Chen et al., 2024), as well as CPT and instruction-tuning recipes (Gao et al., 2025; Xu et al., 2025). Lu et al. (2024) compare context-extension methods under standardized model and data conditions, highlighting the importance of controlled evaluation. Examining learning dynamics, Liang et al. (2026) observe increasing retrieval-head scores during CPT alongside largely stable identities of highly ranked heads in later training. Bertsch et al. (2026) further shows that architectural choices, including normalization and KV sharing, can jointly produce substantial differences in long-context performance that standard short-context metrics poorly predict. Our study addresses how the allocation of attention capacity across heads affects retrieval and adaptation under a common training recipe. We vary head dimension and head count while keeping total projection widths, the query-to-KV head ratio, and per-token KV-cache size fixed.

7 Conclusion

In this paper, we systematically compare language models with identical architectures except for the balance between the number of heads and the dimension of each head, demonstrating that wider heads improve long-context retrieval at fixed model sizes. Our analyses suggest that this improvement is associated with (1) increased parameter association with retrieval heads, as wider-head models have comparable numbers of retrieval heads despite having fewer heads overall, and (2) more effective adaptation of components such as embeddings and MLPs during long-context continual pretraining, rather than attention-map changes alone. The experiments conducted in this paper are limited to the head dimensions of at most 512; further increasing the head dimension requires improvements in efficient attention computation, which is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools for implementing experimental code, brainstorming, collecting related work, and drafting and polishing some parts of the manuscript. We have not used generative AI tools for data synthesis.

REPRODUCIBILITY STATEMENT

We report the full architectural configurations, training hyperparameters, and dataset preprocessing methods in Appendix A. All the datasets used in this work are listed in Table 6.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2305.13245.

NVIDIA : Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, Aleksander Ficek, Alex Kondratenko, Alex Shaposhnikov, Alexander Bukharin, Ali Taghibakhshi, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amy Shen, Andrew Tao, Ann Guan, Anna Shors, Anubhav Mandarwal, Arham Mehta, Arun Venkatesan, Ashton Sharabiani, Ashwath Aithal, Ashwin Poojary, Ayush Dattagupta, Balaram Buddharaju, Banghua Zhu, Barnaby Simkin, Bilal Kartal, Bita Darvish Rouhani, Bobby Chen, Boris Ginsburg, Brandon Norick, Brian Yu, Bryan Catanzaro, Charles Wang, Charlie Truong, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christian Munley, Christopher Parisien, Dan Su, Daniel Afrimi, Daniel Korzekwa, Daniel Rohrer, Daria Gitman, David Mosallanezhad, Deepak Narayanan, Dima Rekesh, Dina Yared, Dmytro Pykhtar, Dong Ahn, Duncan Riach, Eileen Long, Elliott Ning, Eric Chung, Erick Galinkin, Evelina Bakhturina, Gargi Prasad, Gerald Shen, Haifeng Qian, Haim Elisha, Harsh Sharma, Hayley Ross, Helen Ngo, Herman Sahota, Hexin Wang, Hoo Chang Shin, Hua Huang, Iain Cunningham, Igor Gitman, Ivan Moshkov, Jaehun Jung, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jian Zhang, Jiaqi Zeng, Jimmy Zhang, Jinze Xue, Jocelyn Huang, Joey Conway, John Kamalu, Jonathan Cohen, Joseph Jennings, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kezhi Kong, Krzysztof Pawelec, Kumar Anik, Kunlun Li, Kushan Ahmadian, Lawrence McAfee, Laya Sleiman, Leon Derczynski, Luis Vega, Maer Rodrigues de Melo, Makesh Narsimhan Sreedhar, Marcin Chochowski, Mark Cai, Markus Kliegl, Marta Stepniewska-Dziubinska, Matvei Novikov, Mehrzad Samadi, Meredith Price, Meriem Boubdir, Michael Boone, Michael Evans, Michal Bien, Michal Zawalski, Miguel Martinez, Mike Chrzanowski, Mohammad Shoeybi, Mostofa Patwary, Namit Dhameja, Nave Assaf, Negar Habibi, Nidhi Bhatia, Nikki Pope, Nima Tajbakhsh, Nirmal Kumar Juluru, Oleg Rybakov, Oleksii Hrinchuk, Oleksii Kuchaiev, Oluwatobi Olabiyi, Pablo Ribalta, Padmavathy Subramanian, Parth Chadha, Pavlo Molchanov, Peter Dykas, Peter Jin, Piotr Bialecki, Piotr Januszewski, Pradeep Thalasta, Prashant Gaikwad, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Rabeeh Karimi Mahabadi, Rajen Patel, Ran El-Yaniv, Ranjit Rajan, Ria Cheruvu, Rima Shahbazyan, Ritika Borkar, Ritu Gala, Roger Waleffe, Ruoxi Zhang, Russell J. Hewett, Ryan Prenger, Sahil Jain, Samuel Kriman, Sanjeev Satheesh, Saori Kaji, Sarah Yurick, Saurav Muralidharan, Sean Narenthiran, Seonmyeong Bak, Sepehr Sameni, Seungju Han, Shanmugam Ramasamy, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shizhe Diao, Shreya Gopal, Shrimai Prabhumoye, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Siddhartha Jain, Somshubra Majumdar, Soumye Singhal, Stefania Alborghetti, Syeda Nahida Akter, Terry Kong, Tim Moon, Tomasz Hliwiak, Tomer Asida, Tony Wang, Tugrul Konuk, Twinkle Vashishth, Tyler Poon, Udi Karpas, Vahid Noroozi, Venkat Srinivasan, Vijay Korthikanti, Vikram Fugro, Vineeth Kalluru, Vitaly Kurin, Vitaly Lavrukhin, Wasi Uddin Ahmad, Wei Du, Wonmin Byeon, Ximing Lu, Xin Dong, Yashaswi Karnati, Yejin Choi, Yian Zhang, Ying Lin, Yonggan Fu, Yoshi Suhara, Zhen Dong, Zhiyu Li, Zhongbo Zhu, and Zijia Chen. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model, 2025. URL https://arxiv.org/abs/2508.14444.

Amanda Bertsch, Luca Soldaini, Matthew R. Gormley, Graham Neubig, Hannaneh Hajishirzi, Kyle Lo, and Dirk Groeneveld. Cracks in the foundation: Seemingly minor architectural choices impact long context extension. arXiv preprint arXiv:2608.10296, 2026. URL https://arxiv.org/abs/2608.10296. Accepted to COLM 2026.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/2002.07028.

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

Longze Chen, Ziqiang Liu, Wanwei He, Yinhe Zheng, Hao Sun, Yunshui Li, Run Luo, and Min Yang. Long context is not long at all: A prospector of long-dependency data for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8222–8234, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.447. URL https://aclanthology.org/2024.acl-long.447/.

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

Yingqian Cui, Jie Ren, Pengfei He, Jiliang Tang, and Yue Xing. Superiority of multi-head attention in in-context linear regression, 2024. URL https://arxiv.org/abs/2401.17426.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fL4qWkSmtM.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128K context. arXiv preprint arXiv:2402.10171, 2024. URL https://arxiv.org/abs/2402.10171.

Kazuki Fujii, Yukito Tajima, Sakae Mizuki, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Masanari Ohi, Masaki Kawamura, Taishi Nakamura, Takumi Okamoto, Shigeki Ishida, Kakeru Hattori, Youmi Ma, Hiroya Takamura, Rio Yokota, and Naoaki Okazaki. Rewriting pre-training data boosts llm performance in math and code, 2025. URL https://arxiv.org/abs/2505.02881.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7376–7399, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.366. URL https://aclanthology.org/2025.acl-long.366/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco

Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L.

Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Daniel Hsu and Mingyue Xu. Attention-based representations for multi-task computation, 2026. URL https://arxiv.org/abs/2608.04243.

Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. Can perplexity reflect large language model's ability in long text understanding? *arXiv preprint arXiv:2405.06105*, 2024. URL https://arxiv.org/abs/2405.06105.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082.

Yupu Liang, Shuang Chen, Guanwei Zhang, Shaolei Wang, and Suncong Zheng. Revealing the learning dynamics of long-context continual pre-training. arXiv preprint arXiv:2604.02650, 2026. URL https://arxiv.org/abs/2604.02650.

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. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yi Lu, Jing Nathan Yan, Songlin Yang, Justin T. Chiu, Siyu Ren, Fei Yuan, Wenting Zhao, Zhiyong Wu, and Alexander M. Rush. A controlled study on long context extension and generalization in LLMs. arXiv preprint arXiv:2409.12181, 2024. URL https://arxiv.org/abs/2409.12181.

Sadegh Mahdavi, Renjie Liao, and Christos Thrampoulidis. Memorization capacity of multi-head attention in transformers. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.02010.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Krishna C. Puvvada, Faisal Ladhak, Santiago Akle Serrano, Cheng-Ping Hsieh, Shantanu Acharya, Somshubra Majumdar, Fei Jia, Samuel Kriman, Simeng Sun, Dima Rekesh, and Boris Ginsburg. SWAN-GPT: An efficient and scalable approach for long-context language modeling. *arXiv preprint arXiv:2504.08719*, 2025. URL https://arxiv.org/abs/2504.08719v1.

Yifu Qiu, Varun R. Embar, Yizhe Zhang, Navdeep Jaitly, Shay B. Cohen, and Benjamin Han. Eliciting in-context retrieval and reasoning for long-context large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3176–3192, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.165. URL https://aclanthology.org/2025.findings-acl.165/.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference*, 2007. URL https://eurasip.org/Proceedings/Eusipco/Eusipco2007/Papers/a5p-h05.pdf.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, 2017. URL https://arxiv.org/abs/1706.03762.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. Retrieval head mechanistically explains long-context factuality. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=EytBpUGB1Z.

Chejian Xu, Wei Ping, Peng Xu, Zihan Liu, Boxin Wang, Mohammad Shoeybi, Bo Li, and Bryan Catanzaro. From 128K to 4M: Efficient training of ultra-long context large language models. arXiv preprint arXiv:2504.06214, 2025. URL https://arxiv.org/abs/2504.06214.

Peng Xu, Wei Ping, Xianchao Wu, Zihan Liu, Mohammad Shoeybi, and Bryan Catanzaro. Chatqa 2: Bridging the gap to proprietary llms in long context and rag capabilities. *arXiv preprint arXiv:2407.14482*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger

Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=293V3bJbmE.

Penghao Yu, Haotian Jiang, Zeyu Bao, Ruoxi Yu, and Qianxiao Li. The effect of attention head count on transformer approximation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.06662.

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

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

Wuwei Zhang, Fangcong Yin, Howard Yen, Danqi Chen, and Xi Ye. Query-focused retrieval heads improve long-context reasoning and re-ranking. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23791–23805, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1214. URL https://aclanthology.org/2025.emnlp-main.1214/.

A EXPERIMENTAL DETAILS

A.1 ARCHITECTURES

Detailed architectural configurations are shown in Table 5. Models in both scale families are decoder-only Transformers with interleaving sliding window attention (SWA) and full attention (FA) layers. We adopt the standard pre-norm structure and use RMSNorm (Zhang & Sennrich, 2019) for normalization. In all the attention layers, we apply head-wise RMSNorm to each query and key. We also apply RMSNorm to the final representation fed to the unembedding module. Embedding and unembedding have distinct parameters.

We use partial RoPE in full-attention layers, motivated by the proportional RoPE (p-RoPE) design used in Gemma 4 (Team, 2026), which applies RoPE only to a subset of each attention head. Gemma 4 constructs its rotary frequencies relative to the full head dimension and only uses 1/4 of them, effectively retaining the higher-frequency portion of the RoPE spectrum. We use a simpler parameterization, applying RoPE to the first 64 dimensions with a small base $\theta = 100$, which similarly avoids very low-frequency rotary components. The use of high-frequency partial RoPE simplifies the context-extension setup as we do not need any tricks to generalize the model to unseen context length. For SWA layers, we use standard RoPE with $\theta = 10,000$.

A.2 DATASETS

The data mix in each stage is summarized in Table 6. We construct data mix in pretraining and CPT stages with 60% web data from Nemotron-CC-v2, 30% code data from Swallow Code and Swallow Code v2, and 10% SFT-style data from Nemotron-Pretraining-SFT-v1. In our preliminary experiments, mixing code data contributed to the improved performance in retrieval tasks, despite RULER not requiring code comprehension. The data mix used in the SFT stage consists of 50% long sequence SFT data and 50% short sequence chat conversations. All the datasets are tokenized by Qwen3 tokenizer.

Preprocessing for Swallow Code v2: In the original data from Swallow Code v2, one sequence corresponds to one file, whose file path and the repository name are provided as a part of metadata. To enhance training of distant token relations, we concatenate files from the same repository and use the resulting long text as a sequence. We concatenate files in the following format:

```
<repository name="(repository name)">
<file path="(file path)">
(content)
</file>
...
<file path="(file path)">
(content)
</file>
</repository>
```

We only use the resulting repository-wise sequences of length up to 48k for pretraining and those of length ranging between 48k and 384k for CPT.

A.3 TRAINING CONFIGURATION

Table 7 provides the training hyperparameters used in each stage. We use AdamW optimizer in all the stages, and use warmup-stable learning rate scheduling. In the CPT stage, we skip the warm-up phase.

A.4 INFERENCE-TIME ATTENTION SCALING FOR PT MODELS

We evaluate PT checkpoints trained for 100B tokens at a sequence length of 4,096 using position-dependent attention-logit scaling at inference time, without updating any model parameters. The logarithmic scaling has the same functional form as the dynamic attention scaling used for global

Table 5: Architecture configurations.

Model size	0.6B	3B
Vocab size	151,699	151,699
Model dimension	1,024	2,048
Number of layers	24	40
SWA:FA ratio	3:1	3:1
SWA number of heads	4	8
SWA head size	256	256
SWA window size	512	512
FA partial RoPE dimension	64	64
FA partial RoPE θ	100	100
FA KV elements/layer-token (symmetric)	1,024	2,048
FA KV elements/layer-token (asymmetric)	1,280	–
MLP activation function	SwiGLU	SwiGLU
MLP dimension	3,072	6,144

Table 6: Data mix used in each training stage.

Stage	Dataset	Tokens
Pretraining	Nemotron-CC-v2 High Quality (Basant et al., 2025)	60B
	Swallow Code (Fujii et al., 2025)	15B
	Swallow Code v2 (packed by repositories)	15B
	Nemotron-Pretraining-SFT-v1 General	10B
CPT	Nemotron-CC-v2 High Quality	4.8B
	Swallow Code v2 (packed by repositories)	2.4B
	Nemotron-Pretraining-SFT-v1 General	0.8B
SFT	ChatQA2-Long-SFT (Xu et al., 2024)	0.5B
	Nemotron-Instruction-Following-Chat-v1 chat_if	0.5B

Table 7: Training hyperparameters.

	Model size	0.6B	3B
Pretraining	Tokens	100B	100B
	Learning rate	2e-4	1e-4
	LR warm-up steps	250	250
	Weight decay	0.1	0.1
	Batch size	1,024	1,024
	Sequence length	4,096	4,096
CPT	Tokens	8B	8B
	Learning rate	1e-4	5e-5
	LR warm-up steps	0	0
	Weight decay	0.01	0.01
	Batch size	16	16
	Sequence length	262,144	262,144
SFT	Tokens after padding	-	1B
	Learning rate	-	2e-5
	LR warm-up steps	-	50
	Weight decay	-	0.01
	Padded batch size (tokens)	-	1,048,576
	Sequence length after packing	-	32,768
All stages	LM-head z-loss multiplier	1e-4	1e-4

Table 8: Training NTP loss at the final logged interval of PT and Full CPT. Entries for the 0.6B models are means ± sample standard deviations across three runs; entries for the 3B models are single-run values.

Model	PT	Full CPT
0.6B-d128	1.908 ± 0.001	1.911 ± 0.009
0.6B-d256	1.908 ± 0.001	1.888 ± 0.042
0.6B-d512	1.913 ± 0.004	1.912 ± 0.011
3B-d128	1.705	1.677
3B-d256	1.704	1.673
3B-d512	1.700	1.665

NoPE layers in SWAN-GPT (Puvvada et al., 2025); here, we apply it to full-attention layers with partial RoPE.

Let $c > 1$ be a scaling constant and let q_p denote the query vector used to compute attention logits at zero-indexed token position p. For every query head in a full-attention layer, we use

$$a_c(p) = \frac{\log(c+p)}{\log c}, \qquad \widetilde{q}_p = a_c(p) q_p. \tag{1}$$

For a causally visible key k_j at position $j \le p$, this gives

$$\widetilde{z}_{p,j} = \frac{\widetilde{q}_p^\top k_j}{\sqrt{d_k}} = a_c(p) \frac{q_p^\top k_j}{\sqrt{d_k}},$$

where d_k is the query/key dimension.

We choose c to be on the order of the pretraining context length (4,096 tokens) and use $c = 4000$ for all reported scaled-PT evaluations. The scaling is applied only to full-attention layers. We refer to checkpoints evaluated with this procedure as *scaled PT*, labeled "PT (attention scaling)" in the figures.

B Supplementary results on continual pretraining

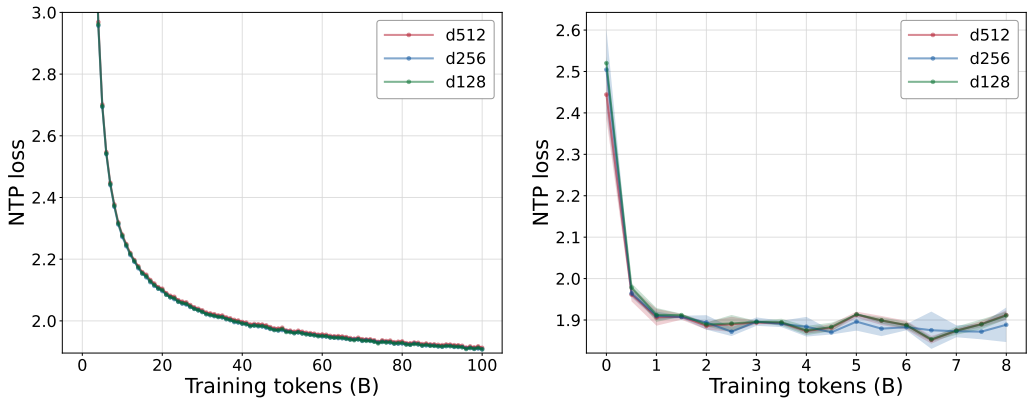
Here we present detailed results for our long-context continual pretraining (CPT) experiments. We first report next-token prediction (NTP) training losses for the 0.6B and 3B models during pretraining (PT) and Full CPT (Appendix B.1). We then present RULER-100 results for both model scales, broken down by task and context length (Appendix B.2). Next, we describe the experimental setup and parameter-update policy for Frozen CPT (Appendix B.3). Finally, we report RULER-100 results by task and context length for the 0.6B models under Frozen CPT and provide detailed comparisons with Full CPT in terms of training loss and RULER-100 performance (Appendix B.4). Unless otherwise specified, CPT refers to Full CPT, in which all model parameters are updated.

B.1 Training loss during PT and Full CPT

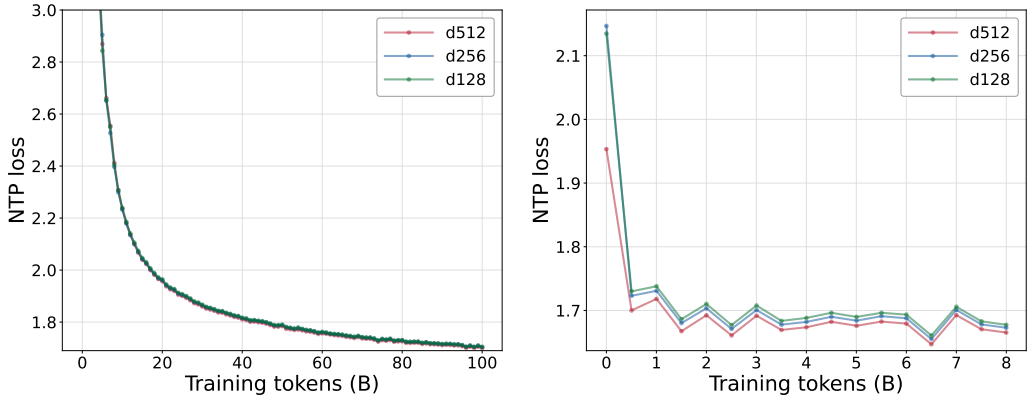
This section provides detailed next-token prediction (NTP) training loss results for the three symmetric head allocations, d128, d256, and d512, at the 0.6B and 3B parameter scales (Table 1).

Figure 7 shows the training NTP loss as a function of the number of training tokens during PT and Full CPT. PT uses 100B tokens at a context length of 4K, followed by an additional 8B tokens of Full CPT at 256K. The horizontal axis counts training tokens in billions within each stage. For the 0.6B models, lines show means across three runs and shaded bands indicate ± 1 sample standard deviation; each 3B head allocation has one run. Table 8 reports the loss at the final logged interval of each stage, with means and sample standard deviations for the 0.6B models and single-run values for the 3B models.

At both model scales, the three head allocations exhibit similar loss trajectories during PT and Full CPT. The final losses also vary little across head dimensions in both stages. Thus, changing head allocation has only a small effect on NTP training loss under our training setup.



(a) 0.6B PT (b) 0.6B Full CPT



(c) 3B PT (d) 3B Full CPT

Figure 7: Training NTP loss during 100B-token PT at 4K and 8B-token Full CPT at 256K. The horizontal axis shows tokens in billions within the indicated stage; the vertical axis shows training cross-entropy. The 3B panels show one run per model. In the 0.6B panels, lines indicate means across three runs and shaded bands indicate ± 1 sample standard deviation.

B.2 RULER-100 RESULTS BY TASK AND CONTEXT LENGTH

Figures 8, 9, and 10 provide task- and length-level RULER-100 results for the 0.6B models after Full CPT, the 3B models after Full CPT, and the 3B models after SFT, respectively. These correspond to panels (a), (b), and (c) of Figure 1. In each figure, the first panel shows the 13-task mean, and the remaining panels show individual task scores across the seven evaluated context lengths from 4K to 256K. For the 0.6B models, thick lines indicate means across three runs and faint lines show individual runs; each 3B model has one run.

The advantage of wider heads is not uniform across individual tasks: a larger head dimension does not always yield a higher score for a given task and context length. These task-level differences may partly reflect run-to-run variability associated with random initialization and other sources of training randomness. Nevertheless, averaging across the 13 RULER-100 tasks reveals a consistent advantage for wider heads: `d512` achieves the highest average score at every evaluated context length in all three settings (Figure 1). Thus, wider heads improve long-context retrieval performance on average across tasks, even though the ordering of head allocations can differ on individual tasks.

Figures 8 and 9 provide the task-level results. Figure 10 reports task-level results for the 3B models after 1B tokens of SFT following CPT. Overall RULER-100 scores increase from 35.93 to 48.28 for `3B-d128`, from 36.72 to 51.81 for `3B-d256`, and from 50.93 to 58.49 for `3B-d512`. The widest-head model therefore retains the highest overall score after SFT. However, scores at 256K decrease from 14.60 to 13.54, from 16.81 to 15.48, and from 23.88 to 20.60, respectively.

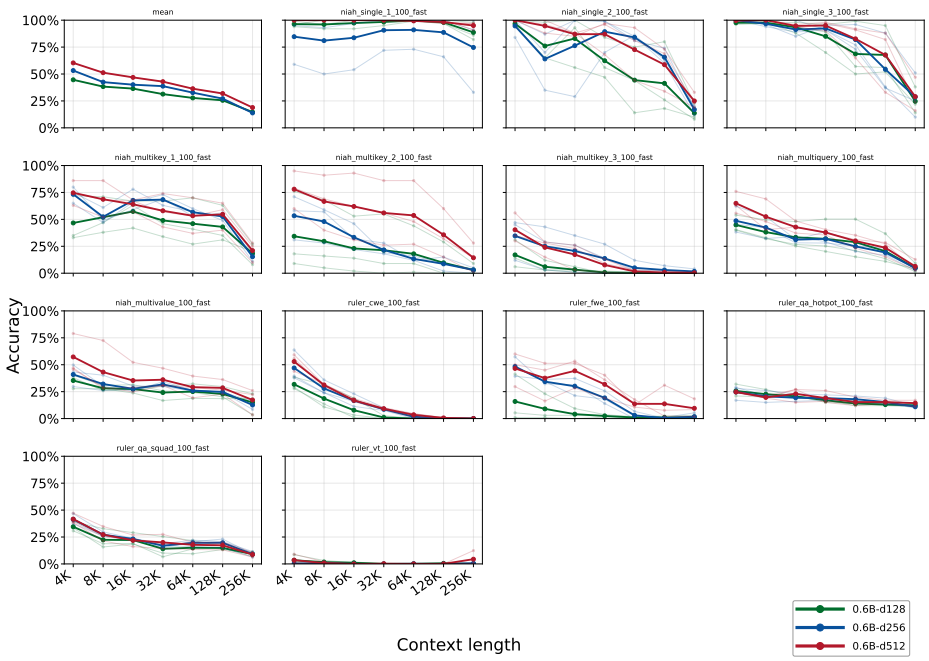


Figure 8: Task-level RULER-100 scores after 8B-token Full CPT for the 0.6B models. The first panel shows the 13-task mean; the remaining panels show individual tasks. Thick lines indicate means across three runs, and faint lines show individual runs. The horizontal axis shows context length, and the vertical axis shows task scores as percentages.

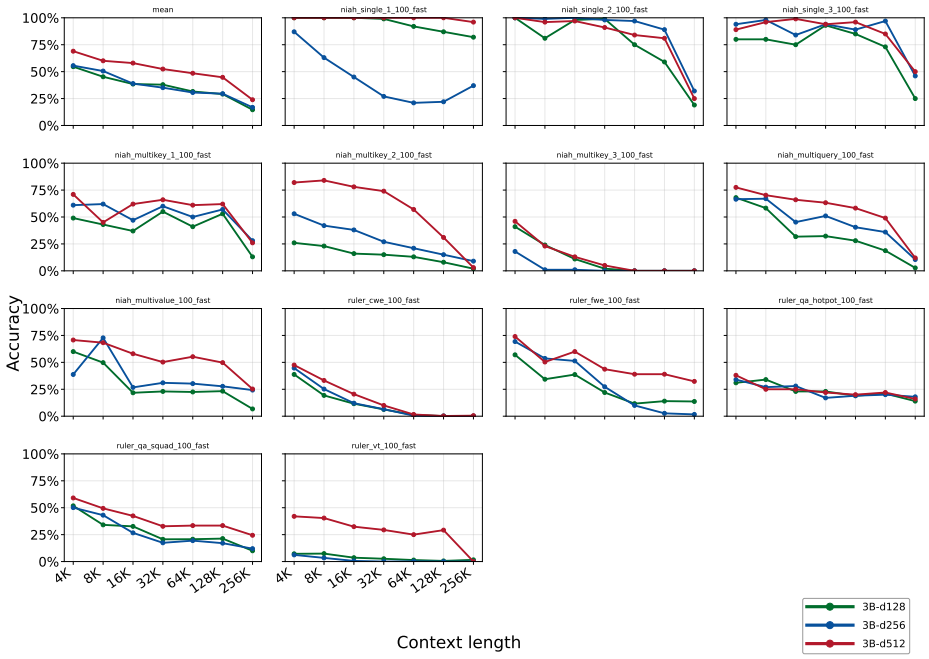


Figure 9: Task-level RULER-100 scores after 8B-token Full CPT for the 3B models. The first panel shows the 13-task mean; the remaining panels show individual tasks. The horizontal axis shows context length, and the vertical axis shows task scores as percentages. Each model has one run. `3B-d512` leads in aggregate but not on every task.

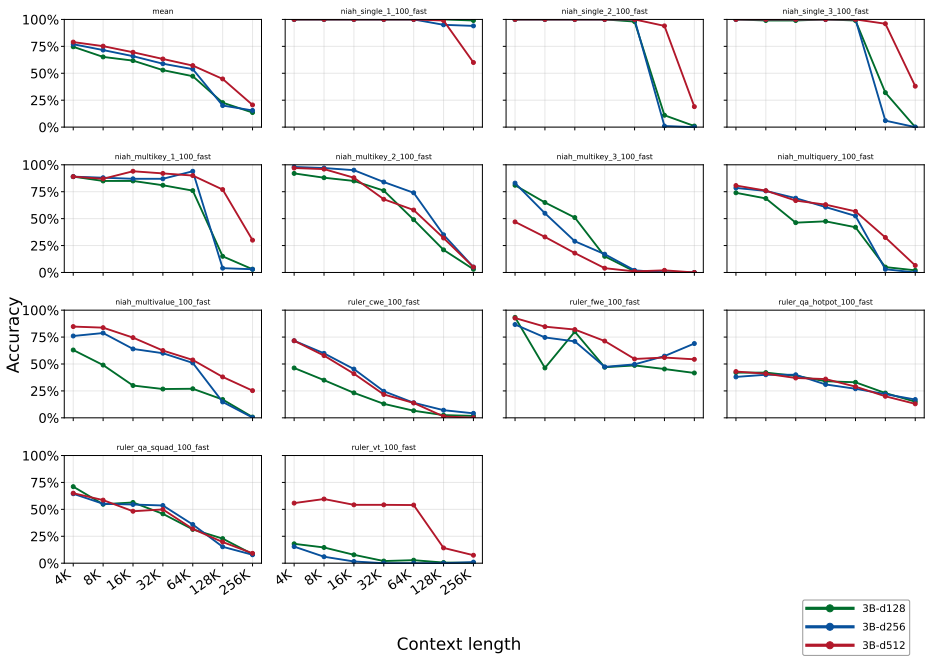


Figure 10: Task-level RULER-100 scores for the 3B models after 1B tokens of SFT following CPT. The first panel shows the 13-task mean; the remaining panels show individual tasks. The horizontal axis shows context length, and the vertical axis shows task scores as percentages. Each model has one run. Relative to the CPT results in Figure 9, overall scores increase, whereas scores at 256K decrease for all three models.

B.3 Experimental setup for Frozen CPT

This section describes the training setup and parameter-update policy for attention-frozen CPT (Frozen CPT). For each of the three symmetric 0.6B configurations, `d128`, `d256`, and `d512`, we compare full-parameter CPT (Full CPT) and Frozen CPT using PT checkpoints from three independent runs. Within each run, both conditions start from the same 100B-token PT checkpoint and undergo 8B tokens of CPT at a context length of 256K.

Full CPT updates all model parameters. In contrast, Frozen CPT keeps the query, key, value, and attention-output projection matrices (W_Q, W_K, W_V, and W_O) in both full-attention and sliding-window attention layers fixed at their PT checkpoint values throughout CPT. We follow the parameter-update policies summarized in Table 9. Token embeddings, the language-model output head, MLP parameters, and all RMSNorm parameters remain trainable.

In particular, the head-wise query/key RMSNorm gain vectors within the attention modules are updated rather than frozen. We keep these gain vectors trainable to allow the attention logit scale to adapt during long-context extension, even when the projection matrices are fixed. Thus, fixed projections do not imply fixed attention maps: both the representations entering attention and the normalization gains can still adapt during Frozen CPT. Neither CPT condition uses the additional inference-time attention scaling applied to the scaled-PT baseline.

B.4 Frozen CPT results and comparisons with Full CPT

This section presents detailed results for the three symmetric 0.6B configurations, `d128`, `d256`, and `d512`, under Frozen CPT and compares them with Full CPT and the corresponding PT checkpoints. For RULER-100 results, we include both unmodified PT and scaled PT, which applies the inference-time attention logit scaling described in Appendix A.4 without updating any model parameters. Neither CPT condition uses this additional scaling.

We first compare the training next-token prediction (NTP) loss during Frozen CPT and Full CPT. Figure 11 shows the loss trajectories as a function of the number of CPT tokens for each head

Table 9: Parameter-update policies during CPT. Fixed parameters retain their values from the paired 100B-token PT checkpoints. Frozen CPT disables updates on weight matrices in both FA and SWA layers, while leaving the listed normalization and non-attention parameters trainable.

Parameter group	Full-Parameter Update (Full)	Attention-Frozen Update (Frozen)
W_Q, W_K, W_V, W_O in FA and SWA layers	Updated	Fixed
Head-wise QK-RMSNorm gain vectors	Updated	Updated
LM head, token embeddings, SwiGLU MLPs, and other RMSNorm parameters	Updated	Updated
Trainable parameter ratio	100%	83.4%

allocation. Table 10 reports the loss at the final logged interval of each condition. The two CPT conditions exhibit similar loss trajectories and final losses, although Frozen CPT attains a slightly lower mean final loss for all three configurations.

We then compare RULER-100 performance across the four conditions. Figure 12 reports mean overall scores, obtained by averaging equally over 13 tasks and seven context lengths from 4K to 256K within each run and then across three runs. Figure 13 provides the corresponding length-wise results, averaging over the 13 tasks within each run and then across three runs.

Relative to unmodified PT, both CPT conditions substantially improve mean overall RULER-100 scores for all three head allocations, with particularly clear gains at longer context lengths. Full CPT achieves higher mean overall scores than Frozen CPT for all three configurations, but the additional benefit is modest and similar across head dimensions: 2.19, 1.85, and 2.32 points for `d128`, `d256`, and `d512`, respectively. Thus, the higher mean retrieval scores under Full CPT are not accompanied by lower mean final training loss.

In contrast, the mean improvement of Frozen CPT over scaled PT depends strongly on head dimension: -0.01, $+3.97$, and $+7.35$ points for `d128`, `d256`, and `d512`, respectively. Frozen CPT therefore nearly matches scaled PT in mean overall performance for `d128`, while achieving progressively larger gains for the wider-head models. The head-dimension dependence of retrieval gains thus persists even without updates to the projection weights in attention modules.

Table 11 reports overall RULER-100 scores for individual training runs. Figures 14, 15, and 16 provide task- and length-level comparisons for the three head allocations. These detailed results show that the relative performance of Full CPT and Frozen CPT can vary across runs, tasks, and context lengths, despite the higher mean overall scores under Full CPT.

C Supplementary results of retrieval-head analysis

This section provides the run-level breakdown of the retrieval-head counts and shared membership summarized in Table 4 and extends the comparison to Frozen CPT. Table 12 reports results for three training runs of each symmetric 0.6B configuration, comparing scaled PT with the corresponding Full CPT and Frozen CPT checkpoints. We use the same detection protocol and retrieval-score threshold across all conditions.

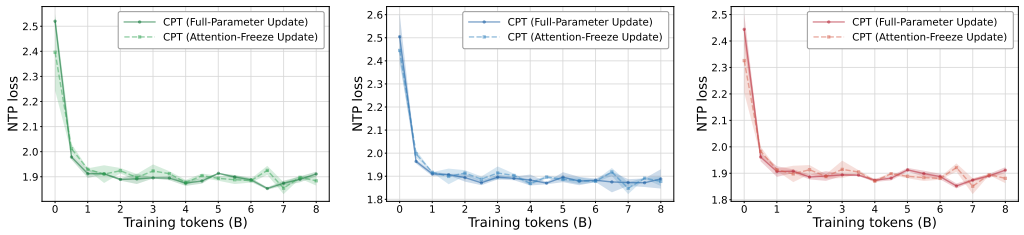
For `d128`, the mean retrieval-head count across three runs increases from 4.00 under scaled PT to 7.00 under both Full CPT and Frozen CPT. Both CPT conditions increase the detected count in all three runs, although the magnitude of the increase varies across runs. Thus, the increase in detected retrieval-head count for `d128` is also observed when the projection weights in attention modules remain fixed during CPT.

Changes relative to scaled PT are smaller for the wider-head models. For `d256`, the mean count is 8.33 under scaled PT, 8.33 under Full CPT, and 8.00 under Frozen CPT. For `d512`, the corresponding mean counts are 5.00, 6.33, and 5.67. These smaller changes are consistent with much of the wider-head models' detected retrieval-head allocation already being visible before CPT under attention scaling.

Full CPT and Frozen CPT yield similar retrieval-head counts across all three configurations. Moreover, most heads detected under Full CPT are also detected under Frozen CPT in each run. These

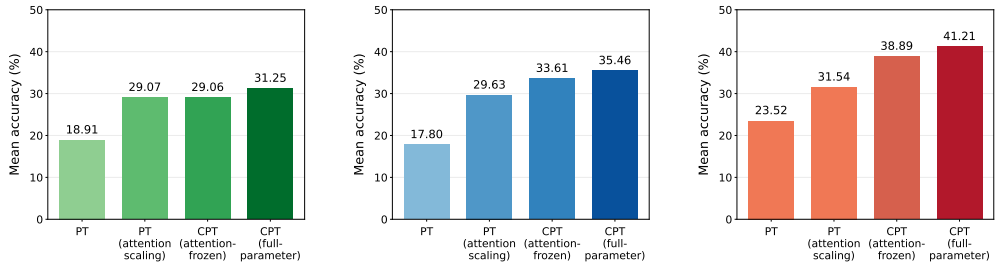
Table 10: Training NTP loss at the final logged interval of Full and Frozen CPT. Entries are means ± sample standard deviations across three runs. Δ_{loss} is the Full-minus-Frozen loss difference, computed within each paired run before aggregation; positive values indicate lower loss under Frozen CPT. Differences are computed before rounding.

Model	Full CPT	Frozen CPT	Δ_{loss}
0.6B-d128	1.911 ± 0.009	1.884 ± 0.019	0.027 ± 0.028
0.6B-d256	1.888 ± 0.042	1.878 ± 0.020	0.011 ± 0.062
0.6B-d512	1.912 ± 0.011	1.881 ± 0.018	0.031 ± 0.028



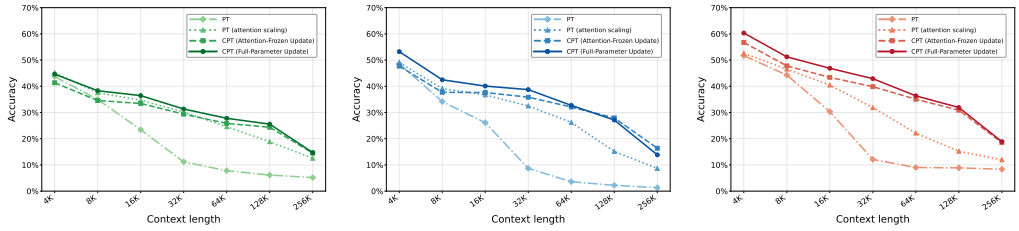
(a) 0.6B-d128 (b) 0.6B-d256 (c) 0.6B-d512

Figure 11: Training next-token prediction (NTP) loss during Full and Frozen CPT for the three 0.6B models. The horizontal axis shows CPT tokens in billions, and the vertical axis shows training cross-entropy. Lines indicate means across three runs, and shaded bands indicate ± sample standard deviation. Frozen CPT attains lower mean final training loss than Full CPT for all three models, despite achieving lower mean overall RULER-100 scores (Figure 4).



(a) 0.6B-d128 (b) 0.6B-d256 (c) 0.6B-d512

Figure 12: Overall RULER-100 scores for unmodified PT, scaled PT, Frozen CPT, and Full CPT in the three 0.6B configurations. Bars show means across three runs, after averaging equally over 13 tasks and seven context lengths from 4K to 256K within each run. Scaled PT uses the inference-time procedure in Appendix A.4; neither CPT condition uses additional scaling.



(a) 0.6B-d128 (b) 0.6B-d256 (c) 0.6B-d512

Figure 13: RULER-100 scores by context length for unmodified PT, scaled PT, Full CPT, and Frozen CPT. At each context length, scores are averaged over 13 tasks within each run and then across three runs. Scaled PT uses the inference-time procedure in Appendix A.4.

Table 11: Overall RULER-100 scores for individual 0.6B training runs, averaged equally over 13 tasks and seven context lengths from 4K to 256K. PT denotes evaluation without additional attention scaling; PT (attention scaling) denotes scaled PT. Full CPT and Frozen CPT follow the parameter-update policies in Table 5. Only Full CPT results are reported for the asymmetric configurations `k512-v128` and `k128-v512`; dashes indicate unreported conditions.

Configuration	Run	PT	PT (attention scaling)	Frozen CPT	Full CPT
d128	1	18.57	28.36	23.23	25.50
d128	2	19.54	31.04	38.13	40.07
d128	3	18.63	27.81	25.83	28.19
d256	1	19.49	30.53	37.24	37.98
d256	2	17.57	29.21	33.81	37.05
d256	3	16.35	29.17	29.77	31.34
d512	1	21.52	24.41	35.43	40.77
d512	2	26.61	36.15	45.58	47.28
d512	3	22.42	34.06	35.65	35.58
k512-v128	1	–	–	–	33.97
k512-v128	2	–	–	–	42.19
k512-v128	3	–	–	–	37.17
k128-v512	1	–	–	–	29.29
k128-v512	2	–	–	–	34.24
k128-v512	3	–	–	–	30.32

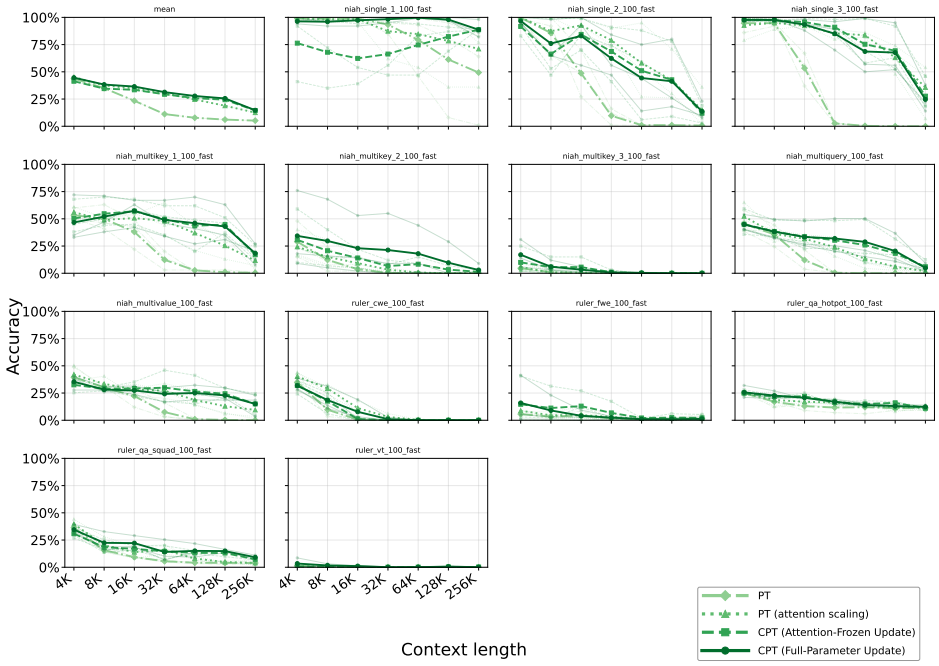


Figure 14: Task-level RULER-100 scores for unmodified PT, scaled PT, Full CPT, and Frozen CPT of `0.6B-d128`. The first panel shows the 13-task mean; the remaining panels show individual tasks. Thick lines indicate means across three runs, and faint lines show individual runs. The horizontal axis shows context length, and the vertical axis shows task scores as percentages. Scaled PT is labeled "PT (attention scaling)" and follows Equation 1.

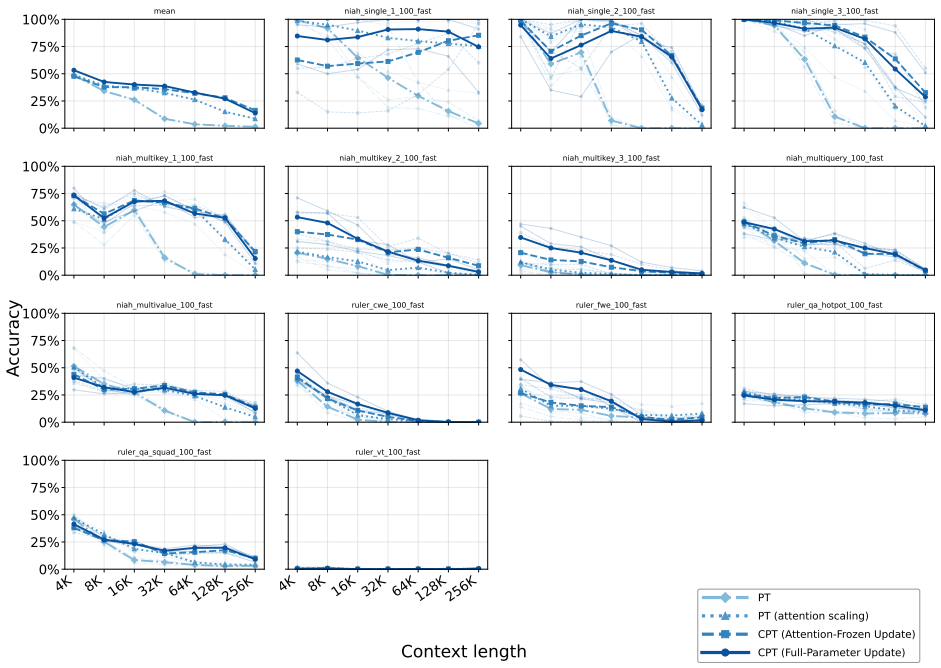


Figure 15: Task-level RULER-100 scores for unmodified PT, scaled PT, Full CPT, and Frozen CPT of `0.6B-d256`. The first panel shows the 13-task mean; the remaining panels show individual tasks. Thick lines indicate means across three runs, and faint lines show individual runs. Axes and scaling follow Figure 14.

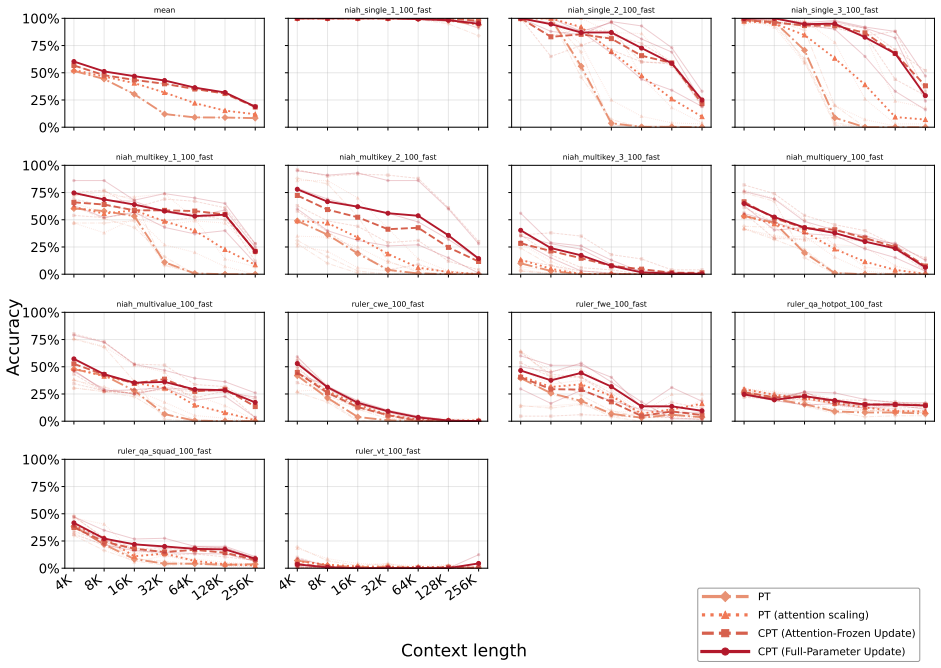


Figure 16: Task-level RULER-100 scores for unmodified PT, scaled PT, Full CPT, and Frozen CPT of `0.6B-d512`. The first panel shows the 13-task mean; the remaining panels show individual tasks. Thick lines indicate means across three runs, and faint lines show individual runs. Axes and scaling follow Figure 14. Full achieves a higher overall mean score than Frozen, with gains and losses depending on the task and context length.

Table 12: Retrieval-head counts and shared membership for each training run. PT denotes scaled PT; Full and Frozen denote the corresponding 8B-token CPT checkpoints without additional scaling. Shared counts are intersections of the detected $(\text{layer}, \text{query head})$ sets. ΔN is the Full-CPT count minus the scaled-PT count; it measures the net count change, not the number of heads newly detected under Full CPT.

Model	Run	Scaled PT	Full CPT	Frozen CPT	Shared PT/Full	ΔN	Shared Full/Frozen
`0.6B-d128`	1	3	9	8	3	+6	8
	2	5	6	7	5	+1	6
	3	4	6	6	4	+2	6
`0.6B-d256`	1	7	7	8	7	0	7
	2	9	9	9	9	0	9
	3	9	9	7	8	0	7
`0.6B-d512`	1	6	9	8	6	+3	8
	2	3	4	3	3	+1	3
	3	6	6	6	6	0	6

results indicate that most of these heads can meet the retrieval criterion after CPT without updates to the projection weights in attention modules.

The scaled-PT results show that much of the retrieval heads detected in the models with wider heads are already visible before CPT under attention scaling. Most heads detected under Full CPT are also detected under Frozen CPT, indicating that updates on projection weights in attention modules during CPT are not necessary for most of these heads to meet the retrieval criterion after training.

D Experimental details for attention-map and representation analyses

This appendix describes the experimental setup and measurement procedures for the analyses in Section 5.2. We first detail the attention entropy and capped top-10k rank-correlation diagnostics. We then describe the residual-stream effective-rank measurements at a context length of 32K.

D.1 Attention entropy and capped rank correlation

This section describes the experimental setup and metrics for the attention-map diagnostics in Figure 5. We analyze all six full-attention (FA) layers of `0.6B-d128`, `0.6B-d256`, and `0.6B-d512`, using three training runs per configuration.

We use six needle-retrieval tasks from RULER—three single-needle variants and three multi-key variants—whose reference answers have a uniquely specified token order. At each context length of 32K and 64K, we use 300 shared examples per length. We teacher-force the reference answers and analyze attention maps from queries predicting the first three reference-answer tokens. These experiments focus on answer initiation, when retrieval is expected to be more demanding because little reference-answer content is available to guide the continuation.

For each combination of an example, an FA layer, a query head, and an answer query, let p_j denote the attention probability at causally visible token position j. We compute attention entropy as

$$H = -\sum_{j \text{ visible}} p_j \log p_j, \tag{2}$$

using natural logarithms. Lower entropy indicates more concentrated attention, but does not establish whether the attended positions are relevant. Comparing the entropy between PT and each CPT condition therefore measures changes in attention probability concentration during CPT.

To measure changes in token prioritization, we compute capped top-10k rank correlation between each CPT checkpoint and the corresponding PT checkpoint. The comparison uses the same example, layer, query-head index, and answer-query position. Let r_j^M denote the descending, one-based rank of the attention probability at position j in checkpoint M, using average ranks for ties. For

corresponding checkpoints A and B, we define

$$\widetilde{r}_j^M(K) = \begin{cases} r_j^M, & r_j^M \le K, \\ K+1, & r_j^M > K, \end{cases} \qquad \rho_K(A,B) = \mathrm{corr}_{\mathrm{Pearson}}\left(\widetilde{\mathbf{r}}^A(K), \widetilde{\mathbf{r}}^B(K)\right) \tag{3}$$

with $K = 10{,}000$ and rank vectors aligned by token position. Capping removes rank distinctions within the low-attention tail while retaining those positions in the correlation. Lower correlation indicates greater changes in which tokens receive high attention and how they are ordered, rather than changes in probability magnitudes.

To guide the choice of the rank cutoff, we examine the cumulative attention probability curves in Figure 17. The figure compares scaled PT, Full CPT at 2B, 4B, and 8B tokens, and Frozen CPT at 8B tokens for the three head allocations at context lengths of 32K and 64K. For each attention distribution, we sort the probabilities in descending order, $p_{(1)} \ge p_{(2)} \ge \cdots$, and define

$$C(n) = \sum_{j=1}^{\min(n,N)} p_{(j)} \tag{4}$$

where N is the number of causally visible positions. Thus, $C(n)$ measures the probability mass assigned to the n most-attended token positions. Sorting and accumulation are performed separately for each attention distribution before aggregation.

As shown in Figure 17, the top 10k positions account for most attention probability mass on average. Guided by these curves, we choose $K = 10{,}000$ as a heuristic cutoff, retaining rank distinctions among higher-attention positions while suppressing those in the low-attention tail.

For Figure 5, both metrics are computed before aggregation. Within each training run and context length, we take an average over examples, FA layers, query heads, and the three selected answer queries. We then report the mean and sample standard deviation across three runs.

D.2 Residual-stream effective rank

This section describes the effective-rank measurements in Figure 6. We analyze the full-attention (FA) blocks of `0.6B-d128`, `0.6B-d256`, and `0.6B-d512` under scaled PT, Frozen CPT, and Full CPT, using the same evaluation examples of 32K context length as in Appendix D.1. Effective rank summarizes the concentration of the singular-value spectrum, with higher values indicating a more evenly distributed spectrum.

We measure the residual stream at three sites within each FA block: immediately before the attention block (Pre-FA), after adding the attention output to the residual stream (Post-FA), and after adding the output of the subsequent MLP block (Post-MLP). For each example and measurement site, we stack the prompt-token representations as rows of a matrix and perform singular value decomposition (SVD). We then average the ordered singular-value spectra across examples and three training runs, separately for each configuration, checkpoint condition, FA block, and measurement site.

Let $\bar{\sigma}_i$ denote the i-th entry of the mean spectrum. We compute effective rank as

$$\mathrm{ER} = \exp\left(-\sum_i p_i \log p_i\right), \qquad p_i = \frac{\bar{\sigma}_i}{\sum_j \bar{\sigma}_j}. \tag{5}$$

Each reported value is therefore the effective rank of the mean singular-value spectrum, rather than the mean of effective ranks computed for individual examples or runs.

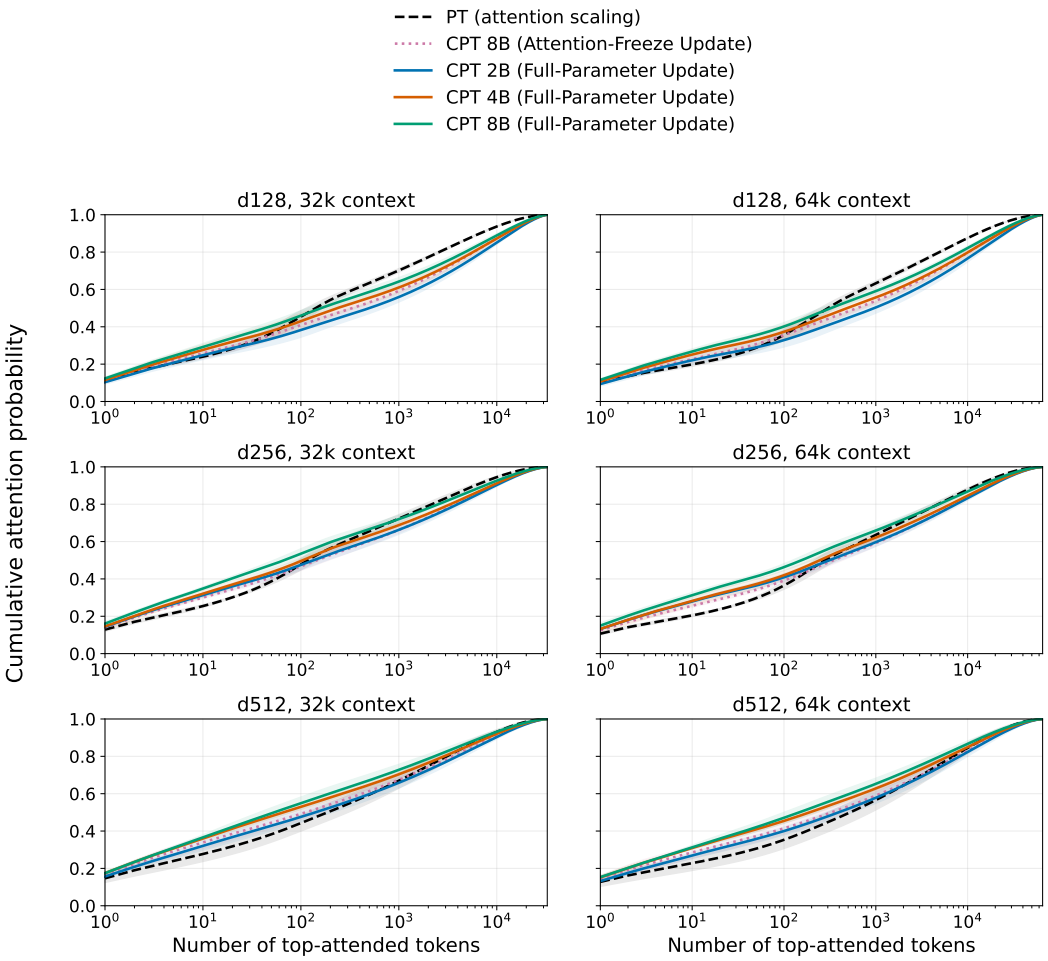


Figure 17: Cumulative attention probability over positions ranked by decreasing attention probability. Rows show `0.6B-d128`, `0.6B-d256`, and `0.6B-d512`; columns show 32K and 64K context lengths. Each panel compares scaled PT, Full CPT at 2B, 4B, and 8B tokens, and Frozen CPT at 8B tokens. The horizontal axis gives the number of ranked positions on a logarithmic scale.