# Rethinking the Role of Efficient Attention in Hybrid Architectures

**Anonymous ACL submission**

## Abstract

Modern language models increasingly adopt hybrid architectures that combine full attention with efficient attention modules, such as sliding-window attention (SWA) and recurrent sequence mixers. However, how these efficient modules shape model capabilities remains poorly understood. To address this gap, we conduct a systematic analysis across hybrid architectures from three perspectives: scaling behavior, mechanism analysis, and architecture design. First, from a scaling perspective, we find that efficient-attention design primarily affects how fast long-context capability emerges, while different hybrids eventually converge to comparable long-context performance under sufficient training. Second, mechanistically, we show that long-range retrieval is mainly carried by full attention, whereas efficient attention shapes its optimization trajectory. This explains a counter-intuitive phenomenon we call *Large-Window Laziness*: larger SWA windows can delay the formation of retrieval heads in full-attention layers. Third, guided by this mechanism, we show that applying NoPE to only the full-attention layers of a small-window SWA hybrid substantially improves long-context performance with negligible impact on short-context performance.

## 1 Introduction

As large language models are increasingly used for long-document understanding and agentic workflows, handling extended contexts has become a core requirement in recent model releases (DeepSeek-AI, 2026; Singh et al., 2025). However, standard softmax attention, which we refer to as *full attention*, is costly at long sequence lengths (Vaswani et al., 2017). This has motivated a family of hybrid attention architectures that combine full attention with *efficient attention* such as sliding-window attention (SWA) (Beltagy et al., 2020) and recurrent sequence mixers (Gu and Dao, 2023; Yang et al., 2024a), a design now widely adopted in recent language models (Agarwal et al., 2025; Gemma Team, 2025; Cao et al., 2026).

Despite their prevalence, the role of efficient attention in hybrid architectures remains unclear. Existing studies lack a unified mechanistic analysis of how different efficient-attention designs shape the capabilities and training dynamics of hybrid architectures, particularly their long-context performance (Xiao et al., 2026; Li et al., 2025; Wang et al., 2025; Bae et al., 2025). To address this gap, we investigate three research questions:

***RQ1 - Scaling Behavior****: How do hybrid architectures scale in short- and long-context performance?*

***RQ2 - Mechanism Analysis****: How does efficient-attention design influence long-context performance?*

***RQ3 - Architecture Design****: What design principles lead to more effective hybrid architectures?*

**Scaling laws for short- and long-context capabilities.** We study how hybrid architectures scale in short- and long-context performance through the lens of *scaling law* across multiple model scales and training budgets (Kaplan et al., 2020; Hoffmann et al., 2022). Considering the discreteness and instability of downstream benchmark scores (Liang et al., 2026), we use validation $\mathrm{Loss}$ and $\log(\mathrm{LongPPL})$ (Fang et al., 2025) as two continuous fitting targets. The former captures general short-context modeling quality, while the latter provides a smooth proxy for long-context capability. The fitted scaling curves clearly show that efficient-attention design has little effect on validation $\mathrm{Loss}$, but leads to more pronounced differences in $\log(\mathrm{LongPPL})$. Specifically, different hybrid architectures exhibit substantial gaps under limited training budgets, with large-window SWA hybrids performing notably worse. However, as training becomes more sufficient, these gaps shrink signifi-

cantly and eventually approach a similar level.

**Efficient attention as an optimization prior.** The scaling pattern above leaves us with two seemingly contradictory puzzles. *First*, why do hybrid architectures with different efficient attention ultimately converge to a similar long-context level? *Second*, why do their convergence rates differ so much, especially across SWA variants with different window sizes? Our mechanistic analysis shows that both puzzles share a common explanation: efficient attention does not directly determine long-context capability; instead, it acts as an *optimization prior* that shapes how full attention is trained.

*Why do hybrids converge?* Through receptive-field constraint and layer-wise probing experiments, we find that long-range information is carried primarily by full attention rather than by efficient-attention modules, even recurrent sequence mixers with in-principle unbounded receptive fields. Sharing this same full-attention component, different hybrids converge to a similar long-context level regardless of their efficient-attention design.

*Why do convergence rates differ?* While full attention sets the final converged level, efficient attention influences long-context capability by shaping how quickly full attention develops its long-range retrieval behavior during training. As a concrete example, by tracing retrieval heads (Wu et al., 2025), we find that retrieval heads form noticeably later in hybrid models equipped with larger SWA windows: once the local window already supplies sufficient context for next-token prediction, the gradient signal pushing full attention to learn long-range retrieval weakens. We term this phenomenon *Large-Window Laziness*.

**Hybrid architecture designs beyond efficient attention.** These findings suggest that hybrid architecture design should focus less on increasing the intrinsic capability of efficient attention and more on helping full attention learn long-range retrieval more effectively. From this perspective, we revisit several design choices beyond the efficient-attention module. As a simple but effective instance, we apply NoPE (Kazemnejad et al., 2023) to the full-attention layers of a small-window SWA hybrid. This simple modification yields a clear long-context capability gain with negligible impact on short-context performance, which is reflected consistently in downstream benchmark evaluations.

Figure 1 summarizes our main findings and their design implications. Taken together, our results reframe the role of efficient attention in hybrid architectures. The practical bottleneck for long-context capability is not simply how powerful the efficient-attention module is, but how it affects the emergence of long-range retrieval in full attention. This view explains the scaling patterns across hybrids and points to full attention as a key target for improving long-context hybrid models.

## 2 Related Work

**Hybrid Attention Architectures.** Existing hybrid architectures mainly follow two lines. One uses SWA (Beltagy et al., 2020) as efficient attention, where recent designs have moved toward smaller windows and sparser full-attention ratios with limited overall performance degradation (Agarwal et al., 2025; Huang et al., 2026). The other employs recurrent sequence mixers that compress past history into a compact recurrent state, such as Lightning Attention (Qin et al., 2024), Mamba-2 (Dao and Gu, 2024), and Gated DeltaNet (Yang et al., 2025b), which are increasingly adopted in recent models (Li et al., 2025; Blakeman et al., 2025; Cao et al., 2026; Team et al., 2026). Beyond the choice of efficient-attention module, recent work also explores head-wise mixing (Dong et al., 2025; Xiao et al., 2025b) and positional encoding for the full-attention layers (Yang et al., 2025a; Puvvada et al., 2025; Chen et al., 2026). However, most of these studies present only final results or limited ablations within specific systems (Gemma Team, 2025; Xiao et al., 2026), leaving a lack of controlled comparisons across efficient-attention architectures.

Several studies have begun to examine structural choices in hybrid architectures more systematically. Wang et al. (2025) compare multiple linear attention variants and mixing ratios, while Waleffe et al. (2024); Bae et al. (2025) analyze layer composition and placement in Mamba-Transformer hybrids. Yet these studies remain within recurrent-mixer-based hybrids and lack a mechanistic explanation. We bridge this gap by comparing different efficient-attention designs under a controlled scaling-law setup and analyzing how they shape the long-context capability of hybrid architectures.

**Scaling Laws and Long-Context Evaluation.** Scaling laws characterize how pretraining performance depends on model and data scale (Kaplan et al., 2020; Hoffmann et al., 2022), with subsequent extensions to transfer learning (Hernan-

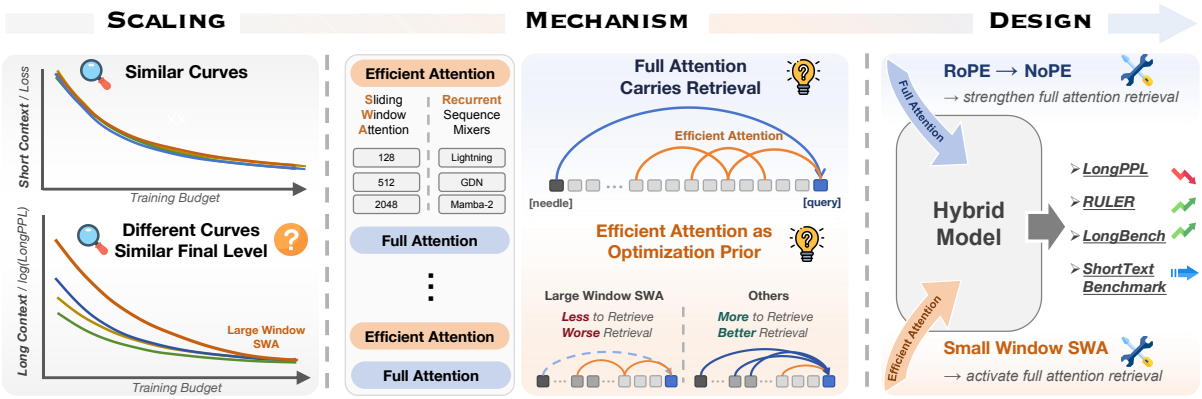



Figure 1: **Overview.** *Scaling*: different efficient-attention designs yield distinct $\log(\text{LongPPL})$ curves that converge to a similar level after sufficient training. *Mechanism*: long-range retrieval is primarily carried by full attention, while efficient attention acts as an *optimization prior*, where large-window SWA lags the most. *Design*: strengthening full attention itself (*e.g.*, RoPE→NoPE in full attention) further improves long-context performance.

dez et al., 2021) and downstream capability prediction (Chen et al., 2024). However, scaling laws for long-context capability remain underexplored. Existing long-context evaluations typically rely on discrete benchmarks such as RULER and LongBench (Hsieh et al., 2024; Bai et al., 2024), which measure final performance but are less suitable for tracking pretraining dynamics. A complementary line of mechanistic studies shows that retrieval heads underlie long-context factual recall (Wu et al., 2025; Xiao et al., 2025a) and tracks the formation of retrieval heads to observe how long-context capability develops during pretraining (Liang et al., 2026), but such signals describe the mechanism rather than quantify capability. In contrast, LongPPL (Fang et al., 2025) provides a continuous perplexity-style metric that correlates strongly with long-context benchmarks, and has since been adopted in recent long-context studies (Song et al., 2026; Willette et al., 2025). We further leverage this metric to fit scaling laws for long-context performance, enabling a more comprehensive comparison of how long-context capability emerges across hybrid architectures.

## 3 Preliminaries

### 3.1 Hybrid Architecture

We cover two common forms of efficient attention: **Sliding-Window Attention (SWA)**, where each token attends only to a finite local window, and **recurrent sequence mixers**, including **Lightning Attention**, **Mamba-2**, and **Gated DeltaNet (GDN)**, which compress past tokens into a recurrent state through different decay strategies and update rules.

We use $q_t, k_t, v_t \in \mathbb{R}^{d_h}$ for the per-head query, key, and value vectors at position $t$ (with $d_k{=}d_v{=}d_h$ assumed for notational simplicity), and let $\mathrm{softmax}_s$ denote the softmax normalized over the index $s$. The formulas below present canonical forms of the mechanisms; implementation-level parameter choices used for matching sizes of different hybrid models are given in Appendix B.

**Full Attention.** For each position $t$, the output $O_t$ is computed over all preceding positions $s \le t$ :

$$O_t = \sum_{s \le t} \mathrm{softmax}_s\big(q_t^\top k_s / \sqrt{d_h}\big)\, v_s \qquad (1)$$

**Sliding Window Attention.** SWA restricts the summation range to a window of size $w$:

$$O_t = \sum_{s \in [t-w+1,\, t]} \mathrm{softmax}_s\big(q_t^\top k_s / \sqrt{d_h}\big)\, v_s \qquad (2)$$

The three recurrent mixers below all share the form $O_t = S_t q_t$ with a recurrent state $S_t \in \mathbb{R}^{d_h \times d_h}$; they differ mainly in how $S_t$ is updated.

**Lightning Attention.** Lightning is a linear attention with a fixed per-head decay $\gamma \in (0, 1)$:

$$S_t = \gamma S_{t-1} + v_t k_t^\top . \qquad (3)$$

**Mamba-2.** Following the structured state-space duality (SSD) form, Mamba-2 can be written as:

$$S_t = \gamma_t S_{t-1} + v_t k_t^\top . \qquad (4)$$

The data-dependent $\gamma_t$ allows per-token control over how much of the past state is preserved.

**Gated DeltaNet.** GDN further adds controlled forgetting through a data-dependent decay $\alpha_t \in (0, 1)$ and a data-dependent update strength $\beta_t \in (0, 1)$:

$$S_t = \alpha_t S_{t-1}(I - \beta_t k_t k_t^\top) + \beta_t v_t k_t^\top. \quad (5)$$

Here, the delta-rule term removes the existing content associated with $k_t$ before writing the new association $v_t k_t^\top$.

### 3.2 Scaling Law

To compare hybrid architectures across model scales and training budgets, we use two fitting targets: validation Loss for short-context modeling and $\log(\text{LongPPL})$ for long-context capability.

**Loss.** Validation loss is the standard target in language-model scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). We select 40K held-out samples from C4 (Raffel et al., 2020) that are disjoint from our training corpus, and report the average negative log-likelihood as Loss.

**LongPPL.** We adopt $\log(\text{LongPPL})$ (Fang et al., 2025) as the fitting target for long-context capability. Following its original implementation, we adopt GovReport (Huang et al., 2021) as the evaluation corpus and Llama-3.1-8B (Grattafiori et al., 2024) as the reference model. More details are provided in Appendix A.

**Scaling Law Formula.** For both $\log(\text{LongPPL})$ and Loss, we model performance as a function of model parameters $N$ (w/o embeddings) and training tokens $D$ (Hoffmann et al., 2022), using the separable power-law form as the fitting template:

$$L(N, D) = aN^{-\alpha} + bD^{-\beta} \quad (6)$$

where $a, b, \alpha, \beta$ are fitted separately for each architecture and fitting target.

## 4 Scaling Behavior of Short- and Long-Context Capabilities

To answer RQ1, we fit scaling laws for validation Loss and $\log(\text{LongPPL})$ to compare short-context and long-context capabilities across hybrid architectures with different efficient-attention designs.

### 4.1 Settings

**Model architecture.** We compare a full-attention Transformer baseline, denoted as *Full*, with six layer-wise hybrid architectures that differ in efficient-attention components. Three hybrids use SWA with window sizes of 128, 512, and 2048, denoted as *SWA-128*, *SWA-512*, and *SWA-2048*. The other three use recurrent sequence mixers, denoted as *Lightning*, *Mamba-2*, and *GDN*. All hybrid models alternate full-attention and efficient-attention layers with a 1:1 ratio.

Table 1: Key hyperparameters of *Full* model for S1–S5.

| Configuration | S1 | S2 | S3 | S4 | S5 |
|---|---|---|---|---|---|
| Params (w/o embed.) | 15M | 31M | 65M | 104M | 477M |
| Total Params | 71M | 107M | 159M | 217M | 665M |
| Layers | 10 | 12 | 16 | 18 | 30 |
| Hidden dim | 384 | 512 | 640 | 768 | 1280 |
| FFN dim | 960 | 1280 | 1600 | 1920 | 3200 |
| Heads (Q) | 6 | 8 | 10 | 12 | 20 |
| Heads (KV) | 2 | 2 | 2 | 2 | 2 |
| Head dim | 64 | 64 | 64 | 64 | 64 |

**Scaling setup.** The scaling study covers five model sizes, S1–S5, with the hyperparameters of the *Full* configuration summarized in Table 1. For the main scaling analysis, we evaluate S1–S4 checkpoints trained with six token budgets, $D \in \{100N, 200N, 300N, 400N, 500N, 1000N\}$, across all architectures, where $N$ corresponds to the parameters without embedding. For S5 scale ($N = 0.48B$ excluding embeddings; total parameters $0.66B$), we train *Full* and *SWA-128* at $D = 100N$ and $D = 200N$ for larger-scale extrapolation checks.

All models are pretrained with a 16K context length on a 1:1 mixture of long and short datasets, which allows us to simultaneously measure short- and long-context capabilities. More training settings are given in Appendix C.

### 4.2 Scaling Law of Validation Loss

We fit the scaling law for validation Loss using 18 data points from S1–S3, and hold out the 6 data points from S4 as a verification set. As shown in Figure 9, all seven architectures are well captured by the scaling law, achieving high $R^2$ on both the fitting and verification sets.

To compare architectures under matched scaling conditions, we examine the predicted Loss at the S5 scale ($N$=0.48B) across training tokens $D$, and include the measured S5 Loss to assess the extrapolation accuracy of the fitted curves.

As shown in the left panel of Figure 2, the validation Loss curves of all hybrid models closely overlap with *Full* across the full range of $D$. This indicates that efficient-attention design has limited impact on short-context capability.

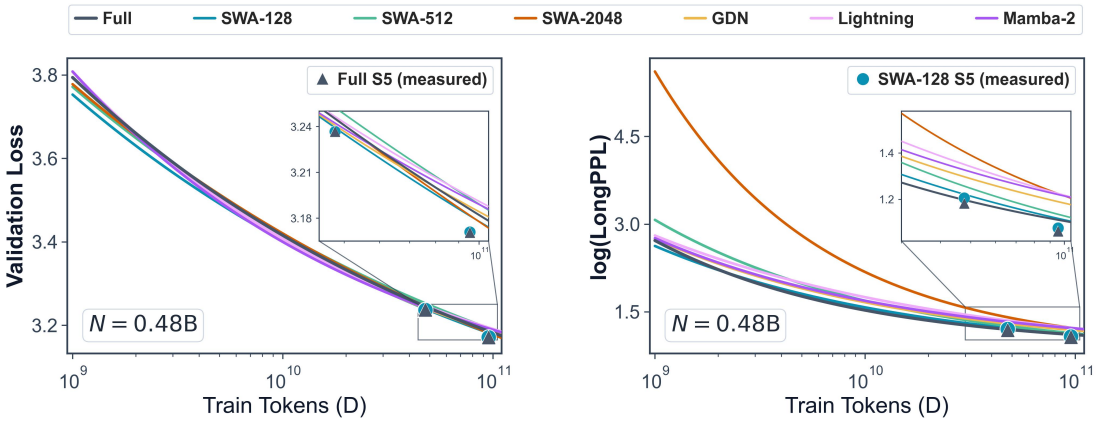



Figure 2: **Predicted** Loss **and** log(LongPPL) **at S5 scale** ($N$=0.48B) **across Train tokens** $D$**.** Loss curves of all hybrids closely overlap, whereas log(LongPPL) curves show large gaps in the low-data regime that shrink with more training. The insets verify extrapolation accuracy against the measured S5 checkpoints of *Full* and *SWA-128*.

### 4.3 Scaling Law of log(LongPPL)

We fit the scaling law for log(LongPPL) following the same protocol as for validation Loss, except that we exclude the S1 checkpoint at $D = 100N$ because its training budget is too small for stable long-context behavior. As shown in Figure 10, although log(LongPPL) is noisier at early checkpoints, it is still smoothly captured by Eq. (6).

In contrast to the strong overlap observed for Loss, the predicted log(LongPPL) reveals much larger architectural differences. We compare architectures under the same setting as above and include the measured S5 log(LongPPL) values to assess extrapolation accuracy.

As shown in the right panel of Figure 2, a clear pattern emerges: architectural differences are most pronounced in early training, corresponding to the low-data regime, where large-window SWA, especially *SWA-2048*, exhibits substantially higher log(LongPPL). As the training becomes more sufficient, this gap rapidly shrinks, and the hybrid models with different efficient-attention designs eventually converge to similar levels with *Full*.

Taken together, the Loss and log(LongPPL) scaling results reveal a clear separation between final capability and training dynamics: **Efficient-attention design has limited effect on the eventual short- and long-context capabilities of hybrid models, but strongly shapes the emergence speed of long-context capability.**

## 5 Mechanism: How Efficient Attention Shapes Long-Context Capability

The key observation in Section 4 naturally motivates ***RQ2:*** *How does efficient-attention design influence long-context performance?* In this section, we conduct a series of mechanistic experiments that dissect the role of efficient attention in long-context modeling; full implementation details and extended analyses can be found in Appendix D.

### 5.1 The Dominant Role of Full Attention

A natural hypothesis is that efficient attention with a larger receptive field, especially recurrent sequence mixers whose receptive field is in principle unbounded, should help improve the long-context capability of hybrid models. However, this is not supported by the scaling pattern that different hybrid models converge to similar log(LongPPL). To examine where long-context capability actually arises, we conduct a receptive-field constraint and a layer-wise probing experiment.

**Receptive-field constraint.** For the S4 models trained with $D = 1000N$ in scaling experiments, we separately restrict the accessible receptive field of efficient attention and full attention to $\approx 2048$ tokens at inference time, and measure the resulting change in log(LongPPL). As shown in Figure 3, when the receptive field of full attention is restricted, log(LongPPL) increases sharply across all hybrid models. In contrast, restricting the receptive field of efficient attention causes only minor changes. This indicates that, in our setting, even recurrent sequence mixers whose receptive field is in principle unbounded and whose update rules are delicate, such as GDN, store little long-range information in their recurrent states during inference.

**Probing Experiment.** To examine how long-range information emerges across layers, we con-

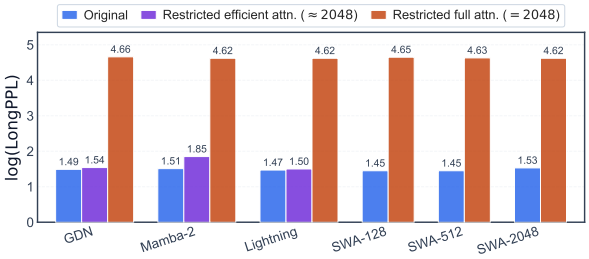



Figure 3: **Inference-time receptive-field restriction for S4/1000$N$ hybrids.** Restricting efficient attention to $\approx$ 2048 tokens leaves $\log(\text{LongPPL})$ nearly unchanged, while restricting full attention raises it sharply.

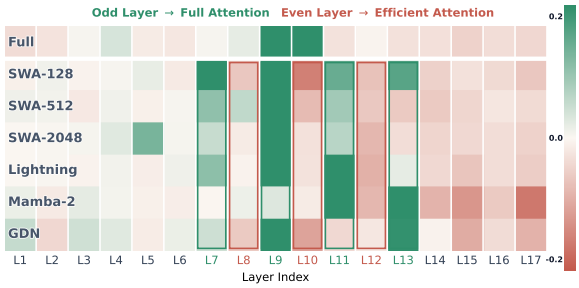



Figure 4: **Layer-wise probing accuracy gain on NIAH for the S4/1000$N$ models.** Cells show incremental accuracy over the previous layer. In all hybrids, gains concentrate at middle full-attention layers (odd-numbered).

duct a layer-wise probing experiment (Belinkov, 2022) on a Needle-in-a-Haystack (NIAH) (Hsieh et al., 2024) classification task. For each layer, we extract the hidden state of the final query token and train a logistic-regression classifier to predict the inserted needle. By comparing the incremental change in probing accuracy from one layer to its predecessor, we estimate how much long-range information is introduced by each layer. Details are provided in Appendix D.2.

Figure 4 shows that, in layer-wise hybrids, probing accuracy increases almost exclusively at middle full-attention layers *(odd-numbered)*, while middle efficient-attention layers *(even-numbered)* contribute little gain and even reduce accuracy. In contrast, *Full* shows continuous growth across middle layers. This supports the view that long-range information in hybrids is mainly introduced and processed by full attention.

The receptive-field constraint and probing experiments suggest that long-context capability in hybrid architectures primarily relies on full attention rather than efficient-attention modules. This also helps explain the scaling behavior observed in Section 4.3: architectural gaps in $\log(\text{LongPPL})$ shrink after sufficient training because all hybrid models share the same full-attention design as *Full*.

### 5.2 Efficient Attention as an Optimization Prior of Long-Context Capability

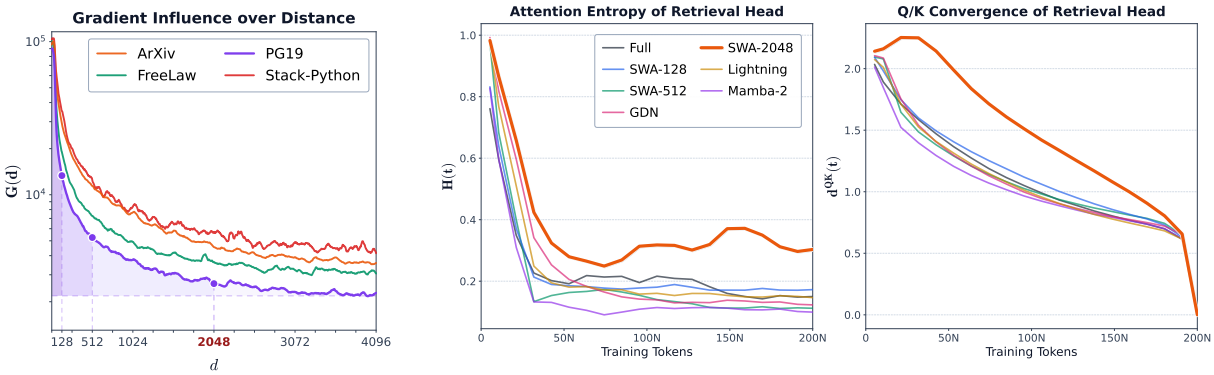
The scaling experiments show that different efficient-attention designs substantially affect the convergence speed of $\log(\text{LongPPL})$. Since long-context capability is primarily carried by full attention, we argue that these differences arise because efficient attention affects how fast full attention learns long-range retrieval. This effect is especially clear in large-window SWA hybrids, which we refer to as *Large-Window Laziness*.

Concretely, a large local window can already cover many useful dependencies during training. As a result, the model can often predict the next token using information within the sliding window, without relying on full attention to retrieve from farther positions. This weakens the optimization pressure for full attention to develop long-range retrieval ability, causing this ability to emerge more slowly. In contrast, SWA with smaller windows leaves more dependencies outside the window range, forcing the model to access them through full attention and thereby providing a denser signal for long-range retrieval. We provide two pieces of evidence consistent with this mechanism.

**Gradient Influence Profiling.** To estimate how next-token-prediction signal decays with distance $d$, we use Llama-3.1-8B to measure the gradient influence $G(d)$ (Li et al., 2016) on long documents sampled from the pretraining corpus. This proxy assumes that the natural long-range dependency distribution is largely model-agnostic, so the profile approximates the dependency signal seen during hybrid-model pretraining. For an input sequence $x_{1:T}$, we define $G(d)$ as

$$G(d) = \mathbb{E}_x \left[ \left\| \frac{\partial s(x)}{\partial e_{T-d}} \right\|_2 \right],$$

where $e_{T-d}$ is the embedding of the token at distance $d$, and $s(x)$ denotes the logit used for prediction. This quantity measures how sensitive the model's prediction is to each historical token, and thus serves as a proxy for distance-dependent signal strength. As shown in Figure 5a, the signal beyond 2048 tokens decays to a flat baseline, while the 512–2048 range still contains substantial gradient signal, which is consistent with *Large-Window Laziness*.




(a) Gradient influence over distance.

(b) Retrieval-head training trajectories.

Figure 5: **Evidence for Large-Window Laziness. (a)** Beyond 2048 tokens, $G(d)$ decays to a flat baseline, while the 512–2048 range still carries substantial signal. **(b)** *SWA-2048* is the outlier: its retrieval-head attention entropy $H(t)$ stays high and Q/K weight distance $d^{\mathrm{QK}}(t)$ shrinks more slowly, indicating under-trained retrieval.

**Retrieval-Head Tracing.** We use retrieval heads (Wu et al., 2025) as the unit of analysis: we densely save intermediate checkpoints before the S4 models reach $D = 200N$ tokens, identify retrieval heads in the final checkpoint, and track two diagnostics at each intermediate checkpoint $t$.

(i) $H(t)$, the normalized attention entropy when retrieving the needle token in the NIAH task:

$$H(t) = -\frac{1}{\log |\mathcal{V}_q|} \sum_{j \in \mathcal{V}_q} a_{qj}^{(t)} \log a_{qj}^{(t)},$$

where $a_{qj}^{(t)}$ is the attention weight from query $q$ to visible key $j$ at checkpoint $t$, and $\mathcal{V}_q$ is the visible-key set. Lower $H(t)$ indicates sharper retrieval.

(ii) $d^{\mathrm{QK}}(t)$, the relative parameter distance from checkpoint $t$ to the final checkpoint:

$$d^{\mathrm{QK}}(t) = \sum_{W \in \{W_Q, W_K\}} \frac{\|W^{(t)} - W^{(t_{end})}\|_F}{\|W^{(t_{end})}\|_F},$$

where $W_Q$ and $W_K$ are the query and key projection matrices of the identified retrieval head, $\|\cdot\|_F$ denotes the Frobenius norm, and $t_{end}$ indexes the $D = 200N$ checkpoint. We report the mean of both diagnostics over the Top-2 retrieval heads.

Figure 5b shows that *SWA-2048* follows a clearly different pattern from the other models: its normalized attention entropy remains high, and its retrieval-head weights converge more slowly, indicating that its retrieval heads remain under-trained. By contrast, retrieval heads train faster in smaller-window SWA and recurrent efficient-attention hybrids, consistent with the need for full attention to access information beyond what the efficient-attention module can reliably provide. We provide additional analyses of retrieval-head formation from complementary perspectives in Appendix D.4, all of which lead to consistent conclusions.

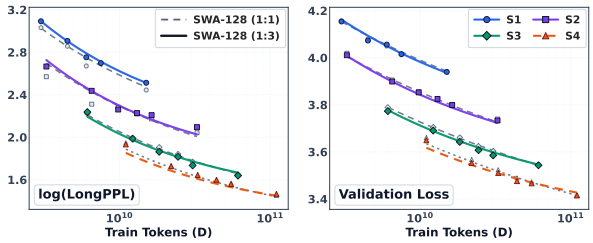



Figure 6: SWA-128 (1:1) vs. SWA-128 (1:3).

Together, these analyses yield a unified mechanistic answer to ***RQ2***: **efficient attention primarily shapes how efficiently full attention learns long-range retrieval, rather than carrying long-range information directly.**

## 6 Hybrid Architecture Design Beyond Efficient Attention

The mechanism above motivates us to revisit hybrid architecture design, raising ***RQ3:*** *What design principles lead to more effective hybrid architectures?* We move beyond efficient attention and examine several other design factors through scaling law and downstream benchmark evaluation.

### 6.1 Full-to-Efficient Layer Ratio

We compare the 1:1 SWA-128 setting used in our main experiments with a sparser 1:3 variant, and fit their validation Loss and log(LongPPL) scaling curves. As shown in Figure 6, the 1:3 ratio gives almost the same validation Loss as the 1:1 ratio. For log(LongPPL), however, the sparser model performs worse at small scales, likely because the number of full-attention layers is too limited. As

Table 2: **Downstream evaluation** of *Full*, *SWA-128*, and *SWA-128-NoPE* at S4 ($0.22B$) and S5 ($0.66B$). $\text{RULER}_{\text{NIAH}}$ is the average over the 8 NIAH-style sub-tasks in RULER; ShortAvg is the average over 19 short-context benchmarks. **Bold** marks the best within each model scale. Full per-task results are reported in Appendix E.

| Setting | Model | Long-Context (32K) | | | Long-Context (16K) | | | ShortAvg |
|---|---|---|---|---|---|---|---|---|
| | | RULER | $\text{RULER}_{\text{NIAH}}$ | LongBench | RULER | $\text{RULER}_{\text{NIAH}}$ | LongBench | |
| $S4(0.22B)$ $D \approx 100B$ | Full | – | – | – | 25.09 | 35.95 | 15.09 | **38.13** |
| | SWA-128 | – | – | – | 35.33 | 49.58 | 15.88 | 38.03 |
| | SWA-128-NoPE | – | – | – | **44.80** | **67.81** | **16.43** | 37.88 |
| $S5(0.66B)$ $D \approx 100B$ | Full | 43.90 | 62.61 | 18.93 | 47.17 | 67.14 | 18.44 | 40.46 |
| | SWA-128 | 41.86 | 60.17 | 18.30 | 46.13 | 65.91 | 17.52 | 41.31 |
| | SWA-128-NoPE | **46.98** | **70.42** | **19.46** | **52.88** | **82.31** | **19.02** | **41.32** |

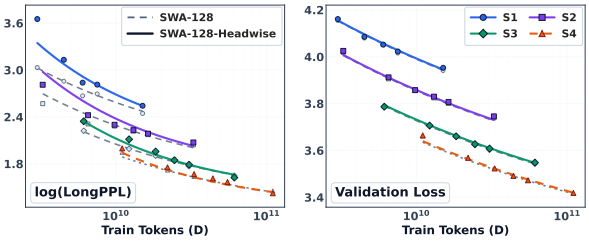



Figure 7: SWA-128 vs. SWA-128-Headwise.

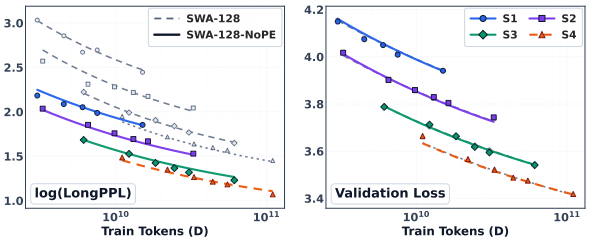



Figure 8: SWA-128 vs. SWA-128-NoPE.

model size increases, this gap closes, suggesting that full-attention density can be safely reduced once enough full-attention layers are available.

### 6.2 Layer-wise vs. Head-wise

Another design choice is whether to place full attention in dedicated layers or distribute it across heads within each layer, as in recent head-wise or intra-layer hybrid designs (Bae et al., 2025). To examine this factor, we compare the layer-wise SWA-128 model with a head-wise variant, SWA-128-Headwise. As shown in Figure 7, under our setting, head-wise mixing does not provide an advantage over layer-wise. Specifically, the two methods reach similar validation Loss and $\log(\text{LongPPL})$ after sufficient training, yet the head-wise variant shows slower $\log(\text{LongPPL})$ convergence.

### 6.3 Positional Encoding of Full Attention

Recent studies show that applying NoPE to full-attention layers can effectively enhance their long-range retrieval capability (Yang et al., 2025a; Puvvada et al., 2025). We take *SWA-128* as the base since it activates full-attention retrieval well, and apply NoPE to its full-attention layers, denoted as *SWA-128-NoPE*. As shown in Figure 8, this change substantially decreases $\log(\text{LongPPL})$ while leaving validation Loss nearly unchanged.

Following the training protocol of scaling experiments, we train *Full*, *SWA-128*, and *SWA-128-NoPE* at S4 (0.22B) and S5 (0.66B) under ≈100B tokens. To further evaluate in longer contexts, we continue training the S5 checkpoints for an additional 5B tokens at a 32K sequence length. For long-context, we use RULER (Hsieh et al., 2024) and LongBench (Bai et al., 2024); for short-context, we report the average over 19 benchmarks. As shown in Table 2, *SWA-128-NoPE* consistently leads on long-context benchmarks at both scales while remaining comparable on short-context tasks.

The design studies suggest that hybrid architecture design should move beyond simply choosing a stronger efficient-attention component and instead prioritize choices that **better activate or directly strengthen full attention, allowing its long-range retrieval capability to emerge more efficiently.**

## 7 Conclusion

Through scaling-law fits and mechanistic analysis, we find that the long-context performance of hybrid models is primarily determined by full attention, while efficient attention, acting as an *optimization prior*, indirectly shapes it by modulating how quickly full attention learns long-range retrieval. This suggests that, under limited training budgets, hybrid design should favor choices that more effectively activate and strengthen the long-context capability of full attention, such as small-window SWA and NoPE, both validated in our experiments.

## Limitations

Although our experiments cover multiple model scales and verify the fitted scaling laws via extrapolation, the largest model we train is still at the sub-billion-parameter level with at most $\approx 100$B pretraining tokens, which is smaller than the scale of frontier industrial systems. We also pretrain directly at a 16K context length and extend to at most 32K, in contrast to the prevailing recipe that pretrains on short context first and subsequently extends to long context. These choices may limit the applicability of our conclusions to larger-scale or differently trained settings.

For efficient-attention designs, we cover representative operators widely adopted in recent hybrid architectures, while leaving out some other popular variants such as RWKV-7 (Peng et al., 2025) and Kimi-Linear (Team et al., 2025). In addition, the design choices discussed in Section 6 are intended to validate our mechanistic conclusions rather than to serve as a full design study, and a more comprehensive verification at larger scales is left to future work.

## References


Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Sangmin Bae, Bilge Acun, Chien-Yu Lin, Haroun Habeeb, Seungyeon Kim, Liang Luo, Junjie Wang, and Carole-Jean Wu. 2025. Hybrid architectures for language models: Systematic analysis and design insights. *arXiv preprint arXiv:2510.04800*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, and 1 others. 2025. Nvidia nemotron 3: Efficient and open intelligence. *arXiv preprint arXiv:2512.20856*.

Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, and 1 others. 2026. Qwen3-coder-next technical report. *arXiv preprint arXiv:2603.00729*.

Yangyi Chen, Binxuan Huang, Yifan Gao, Zhengyang Wang, Jingfeng Yang, and Heng Ji. 2024. Scaling laws for predicting downstream performance in llms. *Transactions on Machine Learning Research*.

Yingfa Chen, Zhen Leng Thai, Zihan Zhou, Zhu Zhang, Xingyu Shen, Shuo Wang, Chaojun Xiao, Xu Han, and Zhiyuan Liu. 2026. Hybrid linear attention done right: Efficient distillation and effective architectures for extremely long contexts. *arXiv preprint arXiv:2601.22156*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, pages 10041–10071. PMLR.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse. In *Proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Mahabaleshwarkar, Shih-Yang Liu, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, and 1 others. 2025. Hymba: A hybrid-head architecture for small language models. In *International Conference on Learning Representations*.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. 2025. What is wrong with perplexity for long-context language modeling? In *International Conference on Learning Representations*.

Gemma Team. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. 2021. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, and 5 others. 2024. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*.

Ailin Huang, Ang Li, Aobo Kong, Bin Wang, Binxing Jiao, Bo Dong, Bojun Wang, Boyu Chen, Brian Li, Buyun Ma, Chang Su, Changxin Miao, Changyi Wan, Chao Lou, Chen Hu, Chen Xu, Chenfeng Yu, Chengting Feng, Chengyuan Yao, and 197 others. 2026. Step 3.5 flash: Open frontier-level intelligence with 11b active parameters. *Preprint*, arXiv:2602.10604.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 conference of the north American chapter of the association for computational linguistics: Human language technologies*, pages 1419–1436.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. 2024. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan.github. io/posts/muon*, 6(3):4.

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

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, and 1 others. 2025. Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691.

Yupu Liang, Shuang Chen, Guanwei Zhang, Shaolei Wang, and Suncong Zheng. 2026. Revealing the learning dynamics of long-context continual pretraining. *arXiv preprint arXiv:2604.02650*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2381–2391.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. 2025. Rwkv-7 "goose" with expressive dynamic state evolution. *Preprint*, arXiv:2503.14456.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273.

Krishna C Puvvada, Faisal Ladhak, Santiago Akle Serrano, Cheng-Ping Hsieh, Shantanu Acharya, Somshubra Majumdar, Fei Jia, Samuel Kriman, Simeng Sun, Dima Rekesh, and 1 others. 2025. Swan-gpt: An efficient and scalable approach for long-context language modeling. *arXiv preprint arXiv:2504.08719*.

Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. 2024. Various lengths, constant speed: Efficient language modeling with lightning attention. In *International Conference on Machine Learning*, pages 41517–41535. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pages 9355–9366. PMLR.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Yunchong Song, Jushi Kai, Liming Lu, Kaixi Qiu, and Zhouhan Lin. 2026. Towards compressive and scalable recurrent memory. *arXiv preprint arXiv:2602.11212*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. 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)*, pages 4149–4158.

Kimi Team, Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, and 1 others. 2025. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*.

MiniCPM Team, Wenhao An, Yingfa Chen, Yewei Fang, Jiayi Li, Xin Li, Yaohui Li, Yishan Li, Yuxuan Li, Biyuan Lin, and 1 others. 2026. Minicpm-sala: Hybridizing sparse and linear attention for efficient long-context modeling. *arXiv preprint arXiv:2602.11761*.

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

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, and 1 others. 2024. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32.

Dustin Wang, Rui-Jie Zhu, Steven Abreu, Yong Shan, Taylor Kergan, Yuqi Pan, Yuhong Chou, Zheng Li, Ge Zhang, Wenhao Huang, and 1 others. 2025. A systematic analysis of hybrid linear attention. *arXiv preprint arXiv:2507.06457*.

Jeffrey Willette, Heejun Lee, and Sung Ju Hwang. 2025. Delta attention: Fast and accurate sparse attention inference by delta correction. *Advances in Neural Information Processing Systems*, 38:12052–12080.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2025. Retrieval head mechanistically explains long-context factuality. In *International Conference on Learning Representations*, volume 2025, pages 62143–62156.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, and 1 others. 2026. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2025a. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. In *International Conference on Learning Representations*, volume 2025, pages 37228–37253.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*.

Liu Xiao, Li Zhiyuan, and Lin Yueyu. 2025b. Wuneng: Hybrid state with attention. *Preprint*, arXiv:2504.19191.

Bowen Yang, Bharat Venkitesh, Dwaraknath Gnaneshwar Talupuru, Hangyu Lin, David Cairuz, Phil Blunsom, and Acyr Locatelli. 2025a. Rope to nope and back again: A new hybrid attention strategy. *Advances in Neural Information Processing Systems*, 38:64133–64157.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2025b. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. 2024a. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning*, pages 56501–56523. PMLR.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. 2024b. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

# A LongPPL Evaluation Details

LongPPL evaluates a model only on tokens whose prediction benefits from long context. Following Fang et al. (2025), we identify these tokens by comparing the token-level negative log-likelihoods assigned by a reference model under full context and under a local chunk. In our experiments, we use GovReport (Huang et al., 2021) as the evaluation corpus and Llama-3.1-8B (Grattafiori et al., 2024) as the reference model.

Let $\ell_{\text{ref}}^{\text{full}}(x_t)$ and $\ell_{\text{ref}}^{\text{chunk}}(x_t)$ denote the token-level negative log-likelihoods assigned by the reference model to token $x_t$ under the full context $x_{<t}$ and under a local chunk, respectively. The set of key tokens is defined as

$$\mathcal{K} = \{\, t : \ell_{\text{ref}}^{\text{chunk}}(x_t) - \ell_{\text{ref}}^{\text{full}}(x_t) > \tau_{\text{gain}}, \quad \ell_{\text{ref}}^{\text{full}}(x_t) < \tau_{\text{nll}} \,\}, \quad \tau_{\text{gain}} = \tau_{\text{nll}} = 2. \tag{7}$$

The first condition selects tokens that receive a clear gain from long context, while the second filters out tokens that remain hard to predict even with full context. For a model $M$ under evaluation, LongPPL is then computed only over $\mathcal{K}$:

$$\text{LongPPL}(M) = \exp\Big(\frac{1}{|\mathcal{K}|} \sum_{t \in \mathcal{K}} \ell_M^{\text{full}}(x_t)\Big). \tag{8}$$

**Evaluation dataset statistics.** Table 3 summarizes the datasets used for the two scaling-law targets. The C4 validation split contains many short documents, making it suitable for measuring short-context modeling quality, whereas the GovReport subset used by LongPPL contains substantially longer sequences and a sufficient number of reference-selected key tokens per example. For GovReport, token lengths are computed after retokenization with the evaluated-model tokenizer and truncation at 16K, which is the pretraining sequence length of our models.

In preliminary experiments, we found that examples with fewer than 10 key tokens often produce unstable LongPPL estimates that occasionally spike to extremely large values. We therefore skip these examples to obtain more stable LongPPL estimates.

# B Model Details

To compare hybrid architectures fairly, we keep the backbone configuration of *Full*, *SWA*, *Lightning*, *Mamba-2*, and *GDN* matched as closely as possible, including the number of layers, hidden size, GQA grouping, and per-head dimension. For efficient attention variants that introduce additional parameters, we make only minimal architectural adjustments so that the total parameter count stays close to the *Full* backbone. This avoids mixing the benefit of extra modules from the original implementations into our comparison of efficient-attention designs.

## B.1 Softmax Attention

For softmax attention, prior work has observed the attention-sink phenomenon, where attention probability can concentrate on a small number of non-semantic positions at the beginning of the sequence (Xiao et al., 2024). To mitigate this, we adopt a learnable per-head softmax sink, as used in recent open models (Agarwal et al., 2025). Concretely, for head $h$ the attention distribution is

$$a_{ij}^{(h)} = \frac{\exp\Big(q_i^{(h)\top} k_j^{(h)} / \sqrt{d_h}\Big)}{\exp(s_h) \;+\; \sum_{\ell \le i} \exp\Big(q_i^{(h)\top} k_\ell^{(h)} / \sqrt{d_h}\Big)},$$

where $s_h$ is a learnable per-head scalar initialized to zero. This is equivalent to introducing a virtual "sink" key with logit $s_h$ that absorbs excess attention mass but contributes nothing to the value aggregation. We enable this sink in all softmax-attention layers.

## B.2 Lightning Attention

Lightning attention is a representative linear-attention variant within the recurrent sequence mixer family introduced in Section 3.1. Compared with a full-attention layer, a Lightning layer introduces a small number of additional parameters. To keep the Lightning hybrid comparable with *Full* and the SWA hybrids in total parameter count, we preserve the GQA configuration, layer count, and backbone hidden size, and only slightly reduce the FFN hidden size inside the Lightning layers. The resulting configuration is summarized in Table 4.

## B.3 Gated DeltaNet

Gated DeltaNet (GDN) is a more elaborate recurrent sequence mixer that combines the gated update of GLA (Yang et al., 2024a) with the delta-rule mechanism of DeltaNet (Schlag et al., 2021; Yang et al., 2024b) (Section 3.1). The standard GDN implementation additionally includes a short 1D convolution on Q/K/V and a value-expansion factor

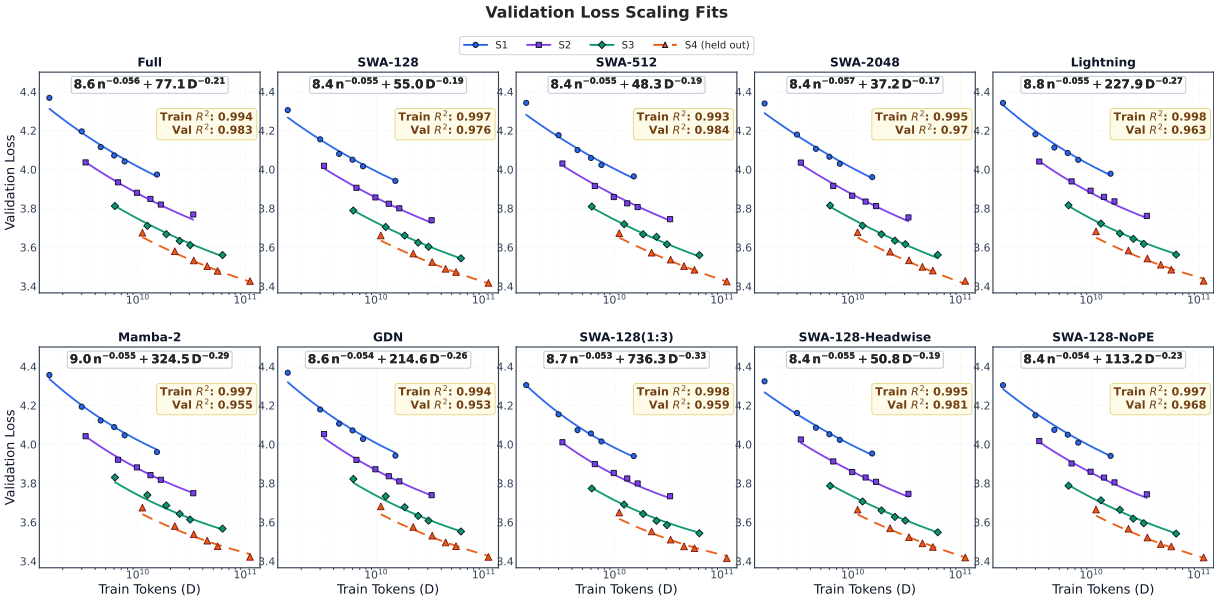



Figure 9: **Validation** Loss **scaling-law fits across all ten architectures.** Each panel plots validation Loss against training tokens $D$ for one architecture at four model scales S1–S4, with the 18 S1–S3 points used for fitting (solid markers) and the 6 S4 points held out for verification (orange triangles). Each colored curve is the fit of $L(N, D) = aN^{-\alpha} + bD^{-\beta}$ (Eq. (6)) at the corresponding $N$, with the fitted coefficients and the train/verification $R^2$ printed inside each panel. The first seven panels cover the architectures studied in the main scaling experiments (Section 4)—*Full* together with three SWA hybrids and three recurrent-mixer hybrids—while the last three panels (*SWA-128(1:3)*, *SWA-128-Headwise*, and *SWA-128-NoPE*) correspond to the design variants from Section 6.

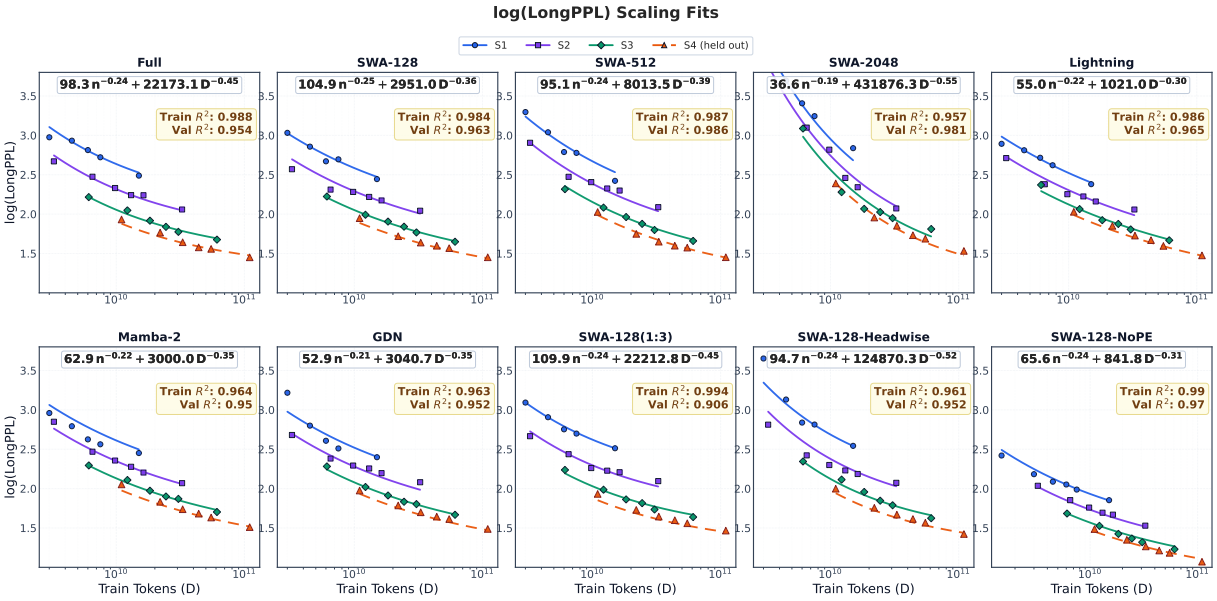



Figure 10: $\log(\text{LongPPL})$ **scaling-law fits across the same ten architectures.** The panel layout, marker convention, and per-panel annotations follow Figure 9. Compared with validation Loss, $\log(\text{LongPPL})$ is noticeably noisier at early checkpoints; in particular, the S1/$D = 100N$ checkpoint is excluded from fitting due to unstable long-context behavior at this small training budget, leaving 17 S1–S3 fitting points and 6 S4 held-out points per architecture. Despite the higher noise level, the same power-law form $L(N, D) = aN^{-\alpha} + bD^{-\beta}$ still fits well.

(expand_v) that widens the value dimension relative to the key dimension. To make the GDN hybrid comparable to *Full* and the other hybrid variants under a matched parameter budget, we make two adjustments to this configuration.

**Removing the short convolution.** The short 1D convolution on Q/K/V is an auxiliary mixing operator not common in Transformer-based models, and

| Dataset / metric | Samples | After filter | Avg tokens | Median tokens | Avg key tokens |
|---|---|---|---|---|---|
| C4 validation / Loss | 40,000 | 40,000 | 497.6 | 269 | – |
| GovReport / LongPPL | 10,000 | 8,898 | 13,317.3 | 13,276 | 78 |

Table 3: Evaluation dataset statistics. Token lengths use the evaluated-model tokenizer; GovReport lengths are after re-tokenization and truncation at 16K. Key tokens are identified by the Llama-3.1-8B reference; "After filter" is the count remaining after dropping examples with fewer than 10 key tokens (no filter is applied to C4).

| Scale | FFN hidden | | Per-layer params | |
|---|---|---|---|---|
| | Full | Lightning | Full | Lightning |
| S1 | 960 | 920 | 1,499,136 | 1,502,208 |
| S2 | 1,280 | 1,240 | 2,621,440 | 2,625,536 |
| S3 | 1,600 | 1,560 | 4,055,040 | 4,060,160 |
| S4 | 1,920 | 1,880 | 5,799,936 | 5,806,080 |

Table 4: Lightning configuration and per-layer parameter counts. Per-layer params report the parameter counts of a single layer; shared LayerNorms and embeddings are omitted.

| Scale | Head Dim | Per-layer params | |
|---|---|---|---|
| | | Full | GDN |
| S1 | 46 | 1,499,136 | 1,496,114 |
| S2 | 46 | 2,621,440 | 2,627,634 |
| S3 | 46 | 4,055,040 | 4,075,570 |
| S4 | 46 | 5,799,936 | 5,839,922 |

Table 5: Gated DeltaNet configuration and per-layer parameter counts. "Head Dim" is the per-group key/value channel dimension $d_k = d_v$ (since expand_v $= 1$); the Full backbone uses $d_k = d_v = 64$.

keeping it would conflate the effect of the recurrent sequence mixer itself with this auxiliary mechanism. We disable it in our main study, and verify with a small ablation at the S1 scale (Table 7). The convolution consistently improves C4 validation loss by a small margin throughout training, but its LongPPL advantage exists only at small training budgets and vanishes once the budget is sufficient. Therefore, disabling the short convolution simplifies the architectural comparison without altering the long-context findings in the paper.

**Keeping FFN width and tuning the state dimension.** At default settings, a GDN layer is heavier than a Full attention layer due to its data-dependent gating and recurrent-state projections. Shrinking the FFN to compensate would require an awkwardly narrow width, so we instead keep the FFN identical to the Full backbone and adjust only the state-related dimensions. We find that expand_v $= 1$ (i.e., $d_v = d_k$) consistently outperforms $d_v > d_k$ on validation loss, and therefore fix expand_v $= 1$ and pick the GDN head dimension so the per-layer parameter count matches *Full*, giving $d_k = d_v = 46$ (Table 5).

| Scale | State Dim | Per-layer params | |
|---|---|---|---|
| | | Full | Mamba-2 |
| S1 | 16 | 1,499,136 | 1,576,098 |
| S2 | 16 | 2,621,440 | 2,789,912 |
| S3 | 16 | 4,055,040 | 4,349,470 |
| S4 | 16 | 5,799,936 | 6,253,092 |

Table 6: Mamba-2 configuration and per-layer parameter counts. State Dim is the SSM state dimension $d_{\text{state}}$. Mamba-2 ends up slightly larger (5%–8%) than Full in order to retain sufficient state capacity.

### B.4 Mamba-2

As with GDN, the standard Mamba-2 implementation includes a short 1D causal convolution before the SSM and an expansion factor expand that widens the SSM hidden dimension. We apply the same strategy as for GDN: disable the convolution to isolate the recurrence, and adjust only the SSM-related dimensions while keeping the FFN unchanged.

A default Mamba-2 layer is also heavier than a Full attention layer, mainly due to the SSM projections ($\Delta_t$, $B$, $C$, $A$) together with the input/output projections. Mamba-2 provides a dedicated state_dim parameter that controls the SSM state size independently of the per-head channel width head_dim, which we use to match the per-layer parameter count to *Full*. Specifically, we set expand $= 1$, keep head_dim $= 64$ to match Full's attention head dimension, and shrink state_dim to 16, which still preserves sufficient state capacity (Table 6).

| **Training tokens** $D$ | $100N$ | $200N$ | $300N$ | $400N$ | $500N$ | $1000N$ |
|---|---|---|---|---|---|---|
| *C4 validation loss (↓)* | | | | | | |
| GDN w/ conv1d | 4.336 | 4.163 | 4.088 | 4.053 | 4.014 | 3.929 |
| GDN w/o conv1d | 4.368 | 4.179 | 4.106 | 4.072 | 4.028 | 3.942 |
| *LongPPL (↓)* | | | | | | |
| GDN w/ conv1d | 80.79 | 19.23 | 15.91 | 13.31 | 12.85 | 11.36 |
| GDN w/o conv1d | 91.35 | 24.96 | 16.44 | 13.56 | 12.30 | 11.01 |

Table 7: Ablation of the short 1D convolution in Gated DeltaNet at the S1 scale. The convolution consistently lowers C4 validation loss by a small margin throughout training. Its LongPPL advantage, however, exists only at small training budgets: the gap closes to within 0.5 by $D{\geq}300N$ and reverses at $D{\geq}500N$.

# C Training Details

For the S1–S4 models, we share the same training hyperparameters (data, sequence length, learning-rate schedule, and batch size) across architectures, so that the scaling comparison is not confounded by optimization or data differences. All models are pretrained with a 16K sequence length, a 1:1 mixture of long and short documents, and a Warmup-Stable-Decay (WSD) learning-rate schedule (Hu et al., 2024). The stable and decay phases account for 90% and 10% of the total training tokens, respectively. During the decay phase, the learning rate is linearly annealed from the stable value to $1/10$ of it. For scaling-law fitting, we use checkpoints at $D/N \in \{100, 200, 300, 400, 500, 1000\}$ for S1–S4 (and $D/N \in \{100, 200\}$ for S5); each checkpoint corresponds to a complete WSD schedule (90% stable plus 10% decay scaled to that $D/N$), not a mid-stable snapshot.

Table 8 summarizes the concrete schedule: S1–S4 are trained to $D/N{=}1000$ while S5 is trained to $D/N{=}200$. The global batch size and stable learning rate at each scale were obtained from a hyperparameter sweep, and we report a configuration that consistently performs well on the Full baseline. For fairness, the same configuration is then shared by all hybrid variants at that scale.

The final row of Table 8 records the long-context extension of the S5/$200N$ checkpoint used in Section 6: we continue training for $\approx 5B$ tokens (4,769 iters) at a 32K sequence length, with the LR linearly decayed from the S5 end LR to 0 (no stable phase) and the RoPE base raised from $10^5$ to $5 \times 10^5$.

# D Mechanism Analysis Details

We conduct several experiments to analyze the mechanism of long-range retrieval in hybrid models, including probing, receptive-field constraints, gradient profiling, and retrieval-head tracing. Here, we provide implementation details for these experiments and explain why they support our conclusions that full attention dominates long-range retrieval and that efficient attention shapes long-context training dynamics by modulating the optimization pressure on full attention.

## D.1 Receptive-field Constraint Details

This section gives implementation details for the inference-time receptive-field restriction experiment in Section 5.1 (Figure 3), where we limit either full attention or efficient attention to a receptive field of $H{\approx}2048$ tokens and measure the change in $\log(\text{LongPPL})$.

**Softmax attention (Full and SWA).** We apply an exact 4D attention mask: for a query at position $i$, attention is allowed only to keys at positions in $[i - H, i]$ with $H = 2048$. This gives a strict per-token receptive field.

**Recurrent kernels (Lightning, Mamba-2, GDN).** The same masking cannot be applied to the recurrent/SSM kernels. We instead use an overlapping-window approximation: the sequence is split into windows of 3072 tokens with a 1024-token stride; within each window, the recurrent state is reset to zero and rolled forward, and only the last 1024 positions of the window are written to the output buffer. Concretely, for a retained block starting at position $s \geq 2048$, the computation window is $[s - 2048,\ s + 1024)$ and the copied-back interval is $[s,\ s + 1024)$, so each token's recurrent state is built from $\approx 2049$ to 3072 preceding tokens. The effective receptive field is therefore slightly looser than the strict $H{=}2048$ used for softmax attention, but is well within the same order of magnitude; we report this as the same "$H{\approx}2048$" condition in Section 5.1.

| Scale | $D/N$ | Global batch | Stable LR | End LR | Stable iters | Decay iters | Total iters | Seq. len. | RoPE base |
|---|---|---|---|---|---|---|---|---|---|
| S1 | 1000 | 32 | $1.953 \times 10^{-3}$ | $1.953 \times 10^{-4}$ | 25,764 | 2,862 | 28,626 | 16K | $10^5$ |
| S2 | 1000 | 16 | $9.766 \times 10^{-4}$ | $9.766 \times 10^{-5}$ | 111,938 | 12,438 | 124,376 | 16K | $10^5$ |
| S3 | 1000 | 28 | $9.542 \times 10^{-4}$ | $9.542 \times 10^{-5}$ | 119,740 | 13,304 | 133,044 | 16K | $10^5$ |
| S4 | 1000 | 64 | $9.766 \times 10^{-4}$ | $9.766 \times 10^{-5}$ | 94,318 | 10,480 | 104,798 | 16K | $10^5$ |
| S5 | 200 | 64 | $9.766 \times 10^{-4}$ | $9.766 \times 10^{-5}$ | 81,900 | 9,100 | 91,000 | 16K | $10^5$ |
| S5 + 32K ext. | – | 32 | $9.766 \times 10^{-5}$ | 0 | 0 | 4,769 | 4,769 | 32K | $5 \times 10^5$ |

Table 8: Training schedule. "$D/N$" is the training budget; iters columns report actual training iters. The final row is the long-context extension used in Section 6. All runs use the Muon optimizer with weight decay 0.1 (Jordan et al., 2024), gradient clipping 1.0.

### D.2 Layer-wise Probing Analysis

This section gives implementation details for the layer-wise probing experiment in Section 5.1 (Figure 4). We probe the S4/1000$N$ checkpoints of seven models: *Full*, *SWA-128*, *SWA-512*, *SWA-2048*, *Lightning*, *Mamba-2*, and *GDN*. The synthetic NIAH classification dataset contains 10,000 samples with a sequence length of 16K and eight candidate classes; its prompt format is illustrated in Figure 11.

For each model and each sample, we run a forward pass with hidden-state output enabled and extract the hidden state of the final query token after every transformer layer. We train an independent logistic-regression probe for each layer, using an 80/20 train/test split with stratified labels and standardizing the hidden states before fitting; the multi-class implementation uses a one-vs-rest scheme. Table 9 shows that logistic regression gives the strongest layer-wise accuracy among the lightweight classifiers we test, so we use it as the primary probe.

Figure 4 visualizes the incremental layer contribution, i.e., the heatmap entries are $A_\ell - A_{\ell-1}$ where $A_\ell$ is the raw probing accuracy at layer $\ell$. Table 10 reports the underlying raw layer-wise accuracies for all 18 layers.

Interestingly, probing accuracy typically peaks at intermediate layers and declines in deeper layers. This suggests that retrieval-related information becomes most linearly accessible in the middle layers, while later layers progressively mix and integrate these signals into higher-level semantic representations, making them less separable by lightweight classifiers. This observation is broadly consistent with prior findings that transformer representations evolve from surface and syntactic features in lower and middle layers toward more abstract semantic representations in deeper layers (Jawahar et al., 2019).

### D.3 Gradient Profiling

Gradient profiling uses the input-gradient norm of a logit-based scalar output as a proxy for the long-range training signal that a historical token provides for next-token prediction. We give a short derivation linking this proxy to (i) local sensitivity of the model's prediction, (ii) gradients on retrieval-head Q/K parameters, and (iii) conditional dependency in the data, and we use it to read Figure 5a.

Let $x_{1:T} \sim \mathcal{D}$ be a token sequence sampled from the pretraining distribution, $e_i \in \mathbb{R}^{d_{\text{model}}}$ the input embedding of $x_i$, and $z^{(t)}(x) \in \mathbb{R}^{|\mathcal{V}|}$ the logit vector produced by $p_\theta$ at position $t$. Following Li et al. (2016), we summarize the model's prediction near the end of the context by the scalar

$$s(x) \;=\; \sum_{v \in \mathcal{V}} \frac{1}{N_\tau} \sum_{t \in \tau} z_v^{(t)}(x),$$

where $\tau$ is the last $N_\tau = 20$ positions, and report the average input-gradient norm at distance $d = T - i$,

$$G(d) \;=\; \mathbb{E}_{x \sim \mathcal{D}}[\|\partial s(x)/\partial e_{T-d}\|_2]\,.$$

**(1) Local sensitivity.** A first-order Taylor expansion of $s$ in $e_i$, followed by Cauchy–Schwarz, gives for any perturbation $\Delta e_i$, up to second-order terms in $\|\Delta e_i\|$,

$$\big|s(e_i + \Delta e_i) - s(e_i)\big| \;\le\; \|\partial s/\partial e_i\|_2 \cdot \|\Delta e_i\|_2.$$

So $\|\partial s/\partial e_i\|_2$ tightly bounds the first-order change of $s$ under infinitesimal perturbations of $e_i$.

**(2) Connection to retrieval-head gradients.** By chain rule, $\partial s/\partial e_i$ decomposes into contributions from all computational paths that route information from position $i$ into the last $N_\tau$ positions. For a single retrieval head with attention weights $a_{t,j}$ and per-position output $o_t = \sum_j a_{t,j} v_j$ (with $v_j = V e_j$), a direct softmax computation gives

$$\frac{\partial s}{\partial \text{score}_{t,i}} \;=\; a_{t,i}\,(v_i - o_t)^\top \frac{\partial s}{\partial o_t},$$

NIAH probing sample format

**Candidate values**: `31415920`, `31415921`, . . . , `31415927`

**Sample fields**:

- Key: `golden-crystal` — sampled adjective–noun identifier
- Value: `31415923` — one value from the eight candidates
- Label: 3 — index of the selected value

**Data**:

A special magic number is hidden within the following text. Make sure to memorize it. I will quiz you about the number afterwards.

*[repeated distractor passage]*

One of the special magic numbers for `golden-crystal` is: `31415923`.

*[repeated distractor passage]*

What is the special magic number for `golden-crystal` mentioned in the provided text?
The special magic number for `golden-crystal` mentioned in the provided text is

Figure 11: Data format for the NIAH classification dataset used in layer-wise probing. The probe predicts the label of the inserted magic number from the final query-token hidden state.

| Classifier | L0 | L1 | L2 | L3 | L4 | L5 | L6 | L7 | L8 | L9 | L10 | L11 | L12 | L13 | L14 | L15 | L16 | L17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Logistic regression | 19.4 | 16.1 | 13.7 | 14.1 | 17.6 | 16.1 | 14.1 | 14.5 | 16.5 | 47.7 | 95.3 | 92.8 | 92.3 | 89.6 | 86.8 | 82.1 | 78.8 | 74.9 |
| MLP | 22.1 | 13.4 | 11.8 | 12.6 | 12.0 | 11.8 | 12.3 | 11.2 | 11.7 | 28.9 | 91.2 | 87.1 | 84.8 | 79.7 | 68.8 | 64.8 | 63.0 | 54.9 |
| Random forest | 30.6 | 15.6 | 13.9 | 13.1 | 14.0 | 12.1 | 12.3 | 11.8 | 12.3 | 11.8 | 18.4 | 18.8 | 16.4 | 15.8 | 14.8 | 14.4 | 14.4 | 13.1 |
| kNN | 20.7 | 14.3 | 13.2 | 12.6 | 12.3 | 12.7 | 11.5 | 12.3 | 12.1 | 12.2 | 13.8 | 13.1 | 13.3 | 13.5 | 12.7 | 12.7 | 12.4 | 12.6 |
| PCA+Naive Bayes | 15.6 | 12.6 | 12.7 | 12.9 | 13.6 | 14.1 | 12.2 | 11.3 | 11.1 | 16.6 | 65.5 | 57.9 | 55.1 | 51.0 | 39.0 | 30.8 | 28.7 | 28.2 |

Table 9: Comparison of lightweight classifiers on the S4/1000$N$ Full model under the same NIAH probing task as Table 10; logistic regression gives the strongest layer-wise accuracy.

so the head's Q/K gradient at the entry $(t, i)$ shares the multiplicative factor $a_{t,i}\, \partial s/\partial o_t$. The same factor also appears in the value-path contribution to $\partial s/\partial e_i$, via $\partial s/\partial v_i = \sum_{t\in\tau} a_{t,i}\, \partial s/\partial o_t$. Hence, absent fine-tuned path cancellation, a small $\|\partial s/\partial e_i\|_2$ implies that the Q/K update strengthening retrieval at distance $d$ is correspondingly weak, and we read $G(d)$ as a per-sample upper-bound proxy on this training signal.

**(3) Connection to data dependency.** If the data satisfies the conditional independence $y_t \perp x_i \mid x_{i+1:t}$ for every $t \in \tau$, then a sufficiently trained $p_\theta$ inherits the same independence in its predictive distribution, and the gradient vanishes:

$$y_t \perp x_i \mid x_{i+1:t} \implies p_\theta(\cdot \mid x_{1:t}) \approx p_\theta(\cdot \mid x_{i+1:t}) \implies \partial s/\partial e_i \approx 0.$$

Conversely, a genuine conditional dependency at distance $d$ forces $\partial s/\partial e_i$ to be nonzero on average. Crucially, $y_t \perp x_i \mid x_{i+1:t}$ is a property of the *data distribution*, so the dependency profile reflected by $G(d)$ transfers across models trained on similar corpora; this justifies using Llama-3.1-8B as a proxy for the dependency signal seen by our hybrid models.[1]

Combining (1)–(3), for a sufficiently trained $p_\theta$, a small $G(d)$ jointly indicates local insensitivity of $s$ to $e_{T-d}$, weak Q/K updates that would strengthen retrieval at distance $d$, and weak conditional dependency at distance $d$ in the data.

[1] Strictly, conditional independence constrains logits only up to a global additive constant (softmax is invariant under such shifts); in the standard parameterization $z_v^{(t)} = w_v^\top h^{(t)}$, this common mode carries no independent training signal.

| Model | L0 | L1 | L2 | L3 | L4 | L5 | L6 | L7 | L8 | L9 | L10 | L11 | L12 | L13 | L14 | L15 | L16 | L17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Full | 19.4 | 16.1 | 13.7 | 14.1 | 17.6 | 16.1 | 14.1 | 14.5 | 16.5 | 47.7 | 95.3 | 92.8 | 92.3 | 89.6 | 86.8 | 82.1 | 78.8 | 74.9 |
| GDN | 19.9 | 25.1 | 21.1 | 25.4 | 28.1 | 26.8 | 28.2 | 32.7 | 27.1 | 74.4 | 63.8 | 60.0 | 58.1 | 77.5 | 77.0 | 67.5 | 64.2 | 55.6 |
| Lightning | 12.1 | 12.2 | 12.4 | 11.9 | 12.7 | 11.6 | 12.8 | 23.5 | 23.0 | 67.0 | 64.5 | 89.1 | 80.2 | 82.2 | 78.4 | 72.0 | 68.8 | 63.4 |
| Mamba-2 | 12.1 | 13.7 | 12.1 | 14.2 | 14.9 | 14.0 | 12.8 | 12.5 | 13.8 | 16.7 | 15.1 | 61.7 | 53.6 | 78.2 | 69.7 | 57.5 | 51.8 | 35.8 |
| SWA-128 | 12.3 | 11.5 | 12.5 | 12.0 | 12.4 | 14.3 | 12.6 | 39.2 | 33.1 | 76.6 | 61.5 | 75.8 | 69.1 | 85.0 | 80.2 | 77.5 | 73.7 | 67.7 |
| SWA-512 | 11.6 | 12.7 | 13.5 | 11.7 | 12.6 | 11.8 | 12.2 | 22.6 | 28.3 | 86.2 | 78.6 | 87.3 | 81.4 | 75.6 | 69.9 | 65.7 | 63.5 | 60.0 |
| SWA-2048 | 12.4 | 12.5 | 12.7 | 13.2 | 15.8 | 28.5 | 29.0 | 34.2 | 32.6 | 69.0 | 66.2 | 72.8 | 64.6 | 61.4 | 57.0 | 53.2 | 50.0 | 45.5 |

Table 10: Layer-wise logistic-regression probing accuracy on the S4/1000$N$ NIAH classification task.

**The flat baseline.** Even when $x_i$ is conditionally uninformative, $G(d)$ does not reach zero in practice; instead, it decays to a flat baseline. Three sources contribute to this irreducible level: (a) finite-precision backward arithmetic, (b) finite-capacity $p_\theta$ that is not exactly Bayes-optimal, and (c) coarse topic/style/domain signals that distant tokens still carry. Formally, even with a mean-zero per-sample gradient, Jensen's inequality gives

$$G(d) \;=\; \mathbb{E}[\|\partial s/\partial e_i\|_2] \;\geq\; \|\mathbb{E}[\partial s/\partial e_i]\|_2,$$

so $G(d)$ remains strictly positive whenever the per-sample gradient is non-degenerate. We therefore estimate the baseline at a distance where Figure 5a has visibly flattened,

$$G_{\text{base}} \;:=\; G_{\text{PG19}}(d = 4096),$$

shown as the dashed reference line in Figure 5a, and treat $G(d) \lesssim G_{\text{base}}$ as effectively no usable retrieval signal at distance $d$. Figure 5a then becomes a quantitative map of the distance ranges that contribute training signal during pretraining, which directly supports the Large-Window Laziness argument in Section 5.2: a SWA window already covering the range where $G(d) \gg G_{\text{base}}$ absorbs most of the dependency-driven training signal before it can propagate to full-attention retrieval heads.

### D.4 Retrieval-Head Tracing

This section gives implementation details for the retrieval-head tracing experiment in Section 5.2 (Figure 5b) and more analysis around the formation of the retrieval head in hybrid architectures.

**NIAH probe and head score.** We construct an NIAH probe where a unique "needle" string is hidden in a long context and the prompt ends with a question whose answer is the needle. Running the S4/200$N$ checkpoint of each hybrid on this prompt, we read the per-head attention from the last input position $q$ (the query) to all keys, and score each head $(\ell, h)$ by the attention mass it places on the needle tokens, averaged over NIAH samples:

$$\overline{\text{score}}_{\ell,h} \;=\; \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \sum_{j \in \mathcal{N}(x)} a_{q,j}^{(\ell,h)}(x),$$

where $a_{q,j}^{(\ell,h)}(x)$ is the attention weight from $q$ to key $j$ in head $(\ell, h)$ for sample $x$, $\mathcal{N}(x)$ is the set of needle token positions, and $\mathcal{S}$ is the NIAH evaluation set. A high $\overline{\text{score}}_{\ell,h}$ means the head consistently routes the query's attention back to the needle—the canonical retrieval-head signature.

**Head selection.** Each cell of Figure 12 reports $\overline{\text{score}}_{\ell,h}$ for one (layer, head), for the six traced hybrid models: *SWA-128*, *SWA-512*, *SWA-2048*, *Lightning*, *Mamba-2*, and *GDN*. We restrict the search to full-attention layers, since our analysis targets long-range retrieval formed there, and select the Top-2 heads per model (red circles) as the retrieval-head set used by the tracing diagnostics in Section 5.2; lower-ranked heads have noisier retrieval signatures and would dilute these diagnostics.

Figure 12 also reveals that *SWA-2048* has noticeably fewer high-response heads in its full-attention layers than the other hybrids, consistent with the *Large-Window Laziness* hypothesis.

**Training Gradient.** To trace the training dynamics of retrieval heads, we train SWA hybrid models with different window sizes and track their gradient norms throughout training. Following the setup of our scaling experiments in Appendix C, we train S1- and S4-scale models from scratch using a constant learning rate for 4000 steps, with an initial 100-step warmup phase. During training, we record the gradients of the $Q$ projection slices for all heads, and use the final checkpoint to identify the Top-1 retrieval head. We then compare the evolution of the gradient norm of this retrieval head

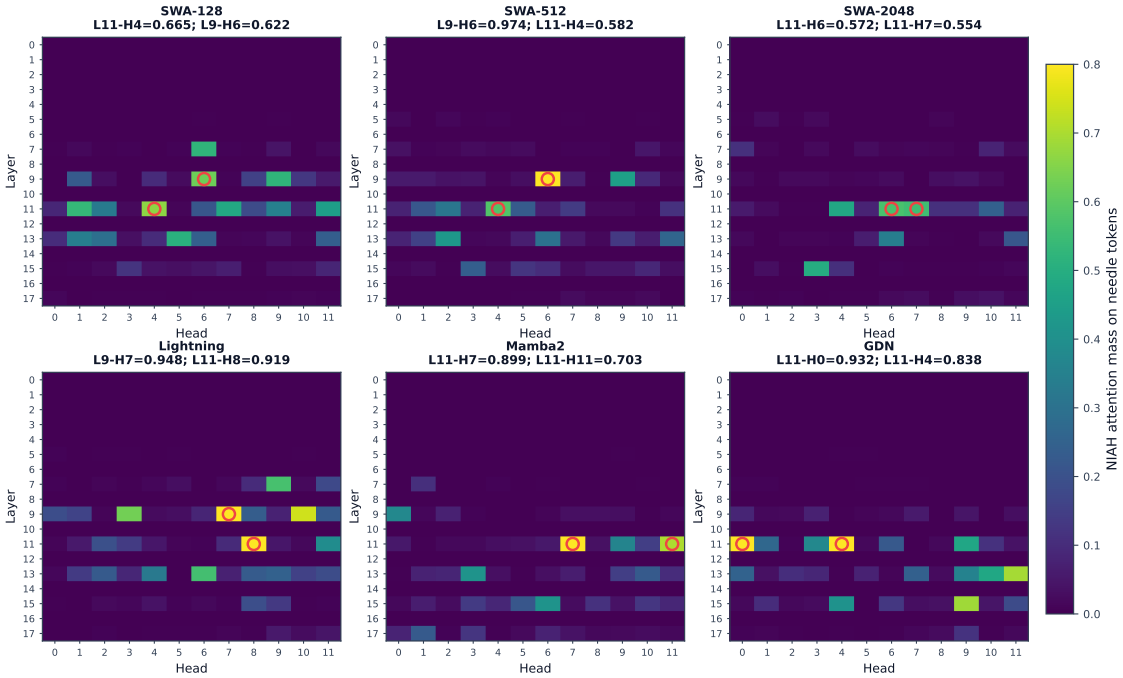



Figure 12: Per-head NIAH attention-mass scores $\overline{\text{score}}_{\ell,h}$ for the six S4/200$N$ hybrid models. Red circles mark the selected top-2 retrieval heads in each model.

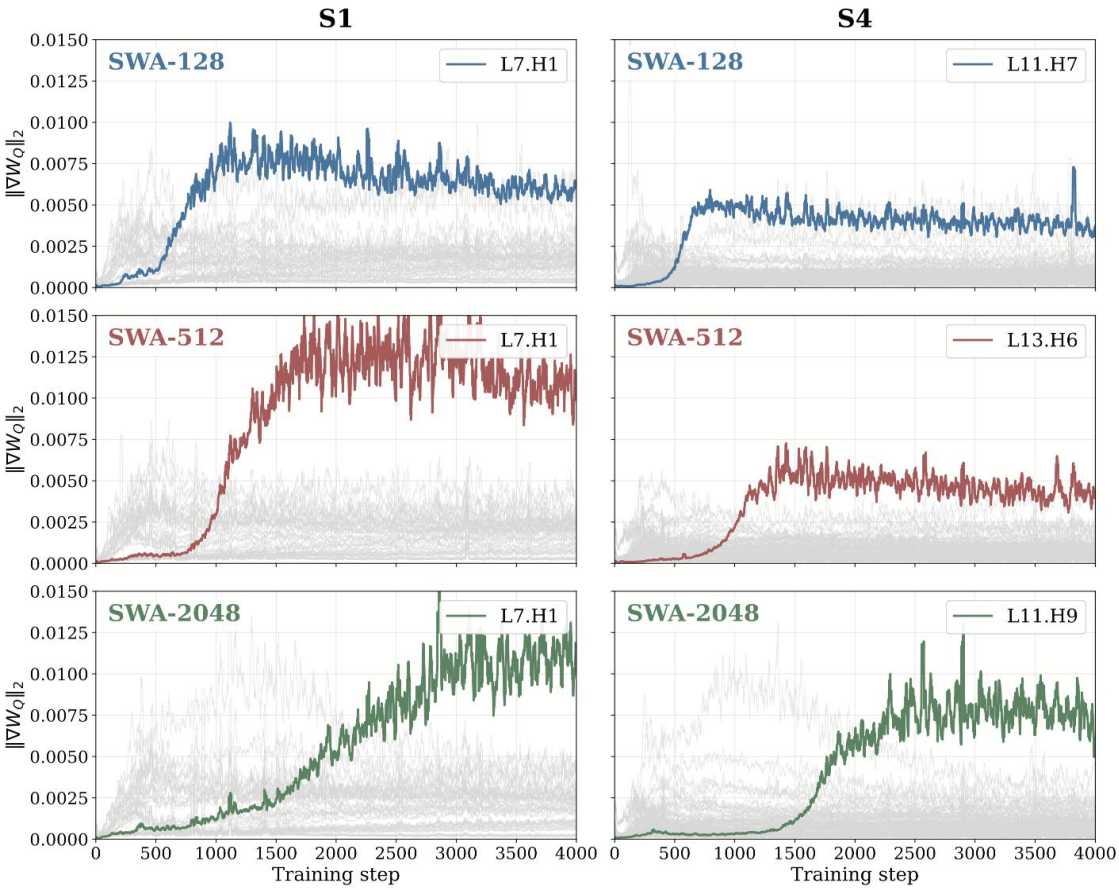



Figure 13: Smaller sliding-window attention activates retrieval-head training earlier. The figure shows the evolution of the Frobenius norm of the gradient on the $Q$ projections of retrieval heads during training for *SWA-128*, *SWA-512*, and *SWA-2048* under both S1 and S4 model scales.

throughout training.

Figure 13 shows the evolution of the Frobenius norm of the loss gradient with respect to the $Q$ projection weights of the retrieval head during training:

$$\|\nabla W\|_F = \left( \sum_{i,j} \left( \frac{\partial \mathcal{L}}{\partial W_{ij}} \right)^2 \right)^{1/2}.$$

We can clearly observe that smaller sliding windows allocate gradient mass to retrieval heads much earlier, whereas larger sliding windows substantially delay the training of retrieval heads. For example, the retrieval head in *SWA-2048* does not begin to receive effective training until roughly 1500 steps into training. The light gray curves in the figure represent the evolution of $\|\nabla W_Q\|_F$ for the other heads in the same model. Notably, for *SWA-2048*, these other heads do not exhibit the same delayed-activation behavior.

## E Benchmark Evaluation

Table 2 in the main paper reports only the aggregated scores. Here we provide the per-task results for the same configurations: *Full*, *SWA-128*, and *SWA-128-NoPE*, each at S4 ($0.22B$) and S5 ($0.66B$) trained under $\approx 100B$ tokens. The 16K-context results use these $\approx 100B$-token checkpoints directly; the 32K-context results use the S5 checkpoint after an additional $5B$-token long-context extension at a 32K sequence length.

**Benchmarks.** For long-context evaluation, we use RULER (Hsieh et al., 2024) and LongBench (Bai et al., 2024); for each RULER sub-task, we generate 200 test instances and report task accuracy averaged over them. For short-context evaluation we use 19 standard benchmarks covering knowledge, commonsense reasoning, reading comprehension and natural language inference: MMLU (Hendrycks et al., 2021), C-Eval (Huang et al., 2023), CMMLU (Li et al., 2024), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), ARC-Easy and ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2021), OpenBookQA (Mihaylov et al., 2018), CommonsenseQA (Talmor et al., 2019), SIQA (Sap et al., 2019), StoryCloze (Mostafazadeh et al., 2016), RACE-middle and RACE-high (Lai et al., 2017), COPA (Roemmele et al., 2011), RTE (Wang et al., 2019), CB (De Marneffe et al., 2019), WiC (Pilehvar and Camacho-Collados, 2019), and MultiRC (Khashabi et al., 2018).

**Evaluation protocol.** All evaluations use deterministic (greedy) decoding to eliminate sampling variance. For long-context tasks, we follow the task-specific reference-based metrics of RULER and LongBench. For short-context multiple-choice tasks, we score each candidate option by its log-likelihood (length-normalized perplexity) under the model and select the option with the highest score; this likelihood-based protocol better reflects the underlying capability of base models, which do not yet have the instruction-following ability needed for direct answer generation.

**Per-task results.** The detailed RULER-16K and LongBench scores are shown in Tables 11 and 12; per-task short-context scores are reported in Table 13.

## F Statement on the AI Usage

During the writing and revision of this paper, the authors used large language models only as auxiliary tools for improving grammar, wording, sentence structure, clarity, and readability. These tools were not involved in the core academic work of this study, including formulating research questions, designing experiments, processing data, analyzing results, or drawing conclusions.

All LLM-assisted edits were carefully reviewed, judged, and revised by the authors. The authors take full responsibility for the authenticity, originality, accuracy, and completeness of the final manuscript.

| Task | S4 / $\approx 100B$ / $16K$ | | | S5 / $\approx 100B$ / $16K$ | | | S5 / $\approx 100B$ / $32K$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | Full | SWA-128 | SWA-128-NoPE | Full | SWA-128 | SWA-128-NoPE | Full | SWA-128 | SWA-128-NoPE |
| *NIAH: single key* | | | | | | | | | |
| niah_s1 | 81.00 | 97.00 | **100.00** | 99.50 | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** |
| niah_s2 | 93.50 | 95.50 | **100.00** | 97.00 | 97.50 | **100.00** | **100.00** | 99.50 | 97.00 |
| niah_s3 | 27.50 | 41.00 | **66.50** | 65.00 | 64.00 | **66.50** | 54.00 | 50.50 | **63.50** |
| *NIAH: multi key* | | | | | | | | | |
| niah_mk1 | 30.00 | 75.50 | **88.00** | **95.50** | 83.50 | 83.50 | **96.50** | 76.00 | 77.50 |
| niah_mk2 | 18.00 | 27.00 | **59.50** | 87.50 | 79.00 | **92.00** | **85.00** | 72.00 | 72.00 |
| niah_mk3 | 1.50 | 3.00 | **27.00** | 19.50 | 17.00 | **48.50** | 5.50 | 17.00 | **30.50** |
| *NIAH: multi value / multi query* | | | | | | | | | |
| niah_mv | 15.25 | 28.38 | **46.50** | 36.25 | 44.12 | **85.50** | 27.88 | 27.12 | **54.62** |
| niah_mq | 20.88 | 29.25 | **55.00** | 36.88 | 42.12 | **82.50** | 32.00 | 39.25 | **68.25** |
| *Variable tracking / aggregation* | | | | | | | | | |
| vt | **1.50** | 0.50 | 0.70 | 3.40 | **4.70** | 0.90 | **8.20** | 6.70 | 3.00 |
| cwe | **4.90** | 1.30 | 2.00 | 0.50 | **7.20** | 4.10 | 0.85 | **3.45** | 2.05 |
| fwe | 12.67 | **36.83** | 14.17 | **40.17** | 35.00 | 0.50 | **44.33** | 37.17 | 14.33 |
| *QA* | | | | | | | | | |
| qa_1 | 14.00 | **16.00** | 15.50 | **20.50** | 18.00 | 15.50 | 8.00 | 8.50 | **17.00** |
| qa_2 | 5.50 | **8.00** | 7.50 | **11.50** | 7.50 | 8.00 | 8.50 | 7.00 | **11.00** |
| *NIAH average (8)* | 35.95 | 49.58 | **67.81** | 67.14 | 65.91 | **82.31** | 62.61 | 60.17 | **70.42** |
| **Total average (13)** | 25.09 | 35.33 | **44.80** | 47.17 | 46.13 | **52.88** | 43.90 | 41.86 | **46.98** |

Table 11: Per-task results on RULER. "NIAH average (8)" is the average over the eight NIAH-style tasks; "Total average (13)" is the average over all 13 RULER tasks.

| Task | S4 / $\approx 100B$ / $16K$ | | | S5 / $\approx 100B$ / $16K$ | | | S5 / $\approx 100B$ / $32K$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | Full | SWA-128 | SWA-128-NoPE | Full | SWA-128 | SWA-128-NoPE | Full | SWA-128 | SWA-128-NoPE |
| *Single-document QA* | | | | | | | | | |
| narrativeqa | **2.82** | 2.52 | 2.53 | 2.72 | 2.93 | **3.10** | 2.74 | 2.87 | **2.88** |
| qasper | 14.07 | **16.46** | 14.39 | 18.72 | 17.09 | **19.07** | 19.32 | 18.30 | **19.78** |
| multifieldqa_en | 16.06 | 17.69 | **17.72** | 19.09 | 19.70 | **21.01** | 20.26 | 20.39 | **21.39** |
| multifieldqa_zh | 13.19 | 13.06 | **13.35** | 17.17 | 16.08 | **17.60** | **17.59** | 15.26 | 15.74 |
| *Multi-document QA* | | | | | | | | | |
| hotpotqa | **6.70** | 6.28 | 6.28 | 7.34 | **8.03** | 7.76 | 7.99 | 8.56 | **8.94** |
| 2wikimqa | **8.36** | 8.12 | 8.17 | 8.69 | 8.18 | **9.27** | 8.58 | 8.44 | **9.77** |
| musique | 3.64 | 3.13 | **3.78** | 3.78 | 4.05 | **5.37** | 4.52 | 4.37 | **5.46** |
| dureader | 18.09 | **19.95** | 18.68 | 25.15 | 22.49 | **26.43** | 23.26 | 23.21 | **25.37** |
| *Summarization* | | | | | | | | | |
| gov_report | 15.33 | 20.60 | **26.76** | 24.46 | **24.48** | 23.40 | 26.15 | **29.54** | 25.68 |
| qmsum | 14.99 | **17.61** | 15.90 | **19.15** | 15.36 | 18.89 | **19.09** | 17.66 | 17.87 |
| multi_news | 17.74 | 19.72 | **25.69** | 17.41 | 22.42 | **23.86** | 21.96 | 25.83 | **26.21** |
| vcsum | 0.90 | 5.65 | **7.51** | 4.34 | **5.18** | 4.39 | 2.14 | **6.54** | 5.04 |
| *Few-shot learning* | | | | | | | | | |
| trec | **71.50** | 62.50 | 67.00 | 69.00 | 66.00 | **71.00** | 65.50 | 66.00 | **71.50** |
| triviaqa | 4.15 | **13.08** | 0.50 | 0.00 | **3.03** | 0.50 | 0.00 | **0.50** | **0.50** |
| samsum | 12.74 | **18.06** | 9.39 | 27.83 | 18.34 | **30.35** | 28.34 | 18.12 | **29.76** |
| lsht | 6.00 | 6.50 | **12.00** | 15.50 | 16.25 | **21.00** | 21.00 | 23.50 | **24.25** |
| *Synthetic* | | | | | | | | | |
| passage_count | **1.98** | 0.33 | 0.23 | 0.97 | 1.17 | **3.13** | **2.35** | 0.62 | 0.96 |
| passage_retrieval_en | 3.83 | 3.71 | **4.01** | **4.17** | 3.67 | 3.83 | 3.54 | **4.79** | 3.88 |
| passage_retrieval_zh | 3.59 | 4.22 | **4.53** | 3.85 | **5.08** | 3.97 | 3.89 | **4.67** | 3.98 |
| *Code completion* | | | | | | | | | |
| lcc | 43.88 | 38.11 | **45.79** | **48.70** | 44.39 | 41.27 | **49.49** | 41.22 | 43.52 |
| repobench-p | 37.33 | 36.16 | **40.78** | **49.24** | 44.06 | 44.14 | **49.88** | 43.84 | 46.09 |
| **Average (21)** | 15.09 | 15.88 | **16.43** | 18.44 | 17.52 | **19.02** | 18.93 | 18.30 | **19.46** |

Table 12: Per-task results on LongBench, using the task-specific reference-based metrics from the official LongBench scripts. The bottom row averages all 21 tasks and matches the LongBench column in Table 2.

| Benchmark | S4 / $\approx 100B$ | | | S5 / $\approx 100B$ | | |
|---|---|---|---|---|---|---|
| | Full | SWA-128 | SWA-128-NoPE | Full | SWA-128 | SWA-128-NoPE |
| *Comprehensive knowledge* | | | | | | |
| MMLU | **26.35** | 24.10 | 25.45 | 29.71 | 30.06 | **30.84** |
| C-Eval | **27.23** | 24.92 | 24.20 | 26.82 | 27.74 | **28.47** |
| CMMLU | **25.63** | 25.18 | 25.12 | 26.20 | **29.46** | 27.48 |
| *Commonsense and completion* | | | | | | |
| HellaSwag | 30.63 | **30.97** | 30.20 | 38.21 | **38.68** | 38.17 |
| PIQA | **64.09** | 61.92 | 61.92 | 65.83 | 65.89 | **66.21** |
| ARC-Easy | 39.51 | 39.15 | **40.04** | 43.03 | **44.80** | 42.15 |
| ARC-Challenge | 23.73 | 25.76 | **28.47** | **31.53** | 30.17 | 28.81 |
| WinoGrande | 52.17 | **53.91** | 52.80 | **54.38** | 53.99 | 53.43 |
| OpenBookQA | **27.60** | 26.60 | 27.40 | 25.00 | 24.80 | **27.60** |
| CommonsenseQA | 19.25 | 19.25 | **19.82** | 21.62 | **24.90** | 22.77 |
| SIQA | 38.08 | **39.41** | 38.08 | 40.63 | **41.50** | 40.58 |
| StoryCloze | **56.97** | 56.87 | 56.60 | 61.73 | 61.57 | **62.27** |
| *Reading and entailment* | | | | | | |
| RACE-middle | **25.07** | 21.87 | 23.12 | 27.51 | 30.15 | **34.75** |
| RACE-high | **25.64** | 21.13 | 21.70 | 28.16 | 29.96 | **30.62** |
| COPA | 51.00 | **54.00** | 49.00 | **58.00** | 56.00 | 57.00 |
| RTE | 48.74 | **53.43** | 53.07 | **52.71** | 51.62 | 50.54 |
| CB | **50.00** | **50.00** | **50.00** | 44.64 | **50.00** | **50.00** |
| WiC | **50.00** | **50.00** | **50.00** | **50.00** | **50.00** | **50.00** |
| MultiRC | 42.82 | **44.08** | 42.80 | 43.09 | **43.67** | 43.34 |
| **Average (19)** | **38.13** | 38.03 | 37.88 | 40.46 | 41.31 | **41.32** |

Table 13: Per-task results on the 19 short-context benchmarks; bottom row averages all 19 tasks and matches the ShortAvg column in Table 2. MMLU, C-Eval, and CMMLU report macro averages over their sub-tasks; the remaining rows report individual benchmark accuracies. All scores are obtained with deterministic decoding and option selection by length-normalized log-likelihood (higher is better).