WHEN SHORTER IS BETTER: VERY SHORT SLIDING WINDOWS IN GATED HYBRID ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

As modern applications demand larger models and longer contexts, global attention has become a dominant cost in both training and inference. Hybrid attention that combine sliding-window attention with global attention reduce this cost, but are typically viewed as trading quality for efficiency, with larger windows assumed to be better. In this paper, we revisit this view through the lens of recent findings on pretrained LLMs, in particular the literature on attention sinks. We find that hybrid architectures with very short sliding windows (16 or 32 tokens) outperform both the full global attention baseline and hybrids with larger windows, even though the latter are strictly more expressive. Adding conditional gating that dynamically balances head contributions further amplifies these gains. We validate both findings with ablations over head configurations, architectural variants, sequence lengths, and training budgets. Our results challenge the conventional trade-off between window size and quality, suggesting that short windows act as a beneficial inductive bias rather than a cheap approximation to global attention.

1 INTRODUCTION

The success of Transformers is often attributed to the expressiveness of global attention (Vaswani et al., 2017). By allowing each token to attend directly to any visible position, global attention makes almost no assumptions about the structure of the input and leaves the model to learn which relationships most useful for the task. This expressiveness is powerful but expensive: compute grows quadratically with sequence length and the key-value cache grows linearly, making attention a major bottleneck for long-context tasks (Dao, 2024; Kwon et al., 2023).

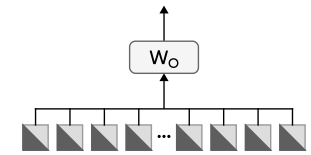
Hybrid attention trades some of this expressiveness for efficiency by imposing a locality prior: it keeps global attention in a subset of layers or heads and replaces the rest with sliding-window attention (SWA). Several recent model families adopt this design with windows of hundreds to thousands of tokens (Gemma Team et al., 2024; Gemma Team, 2025; OpenAI, 2025; Olmo Team et al., 2025). Such designs can improve efficiency with little loss in quality, though the tradeoff depends on the window size and on how global attention is allocated (Gemma Team, 2025; Wang et al., 2025).

In this paper, we ask a different question: since we are imposing a locality prior anyway, can the right prior make the model *better*, not merely cheaper?

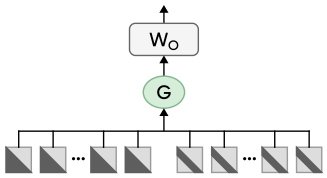
Locality is well motivated by studies of the internal workings of pretrained large language models (LLMs). Beginning with StreamingLLM (Xiao et al., 2024), a line of work has shown that most heads attend almost exclusively to a few initial tokens (attention sinks) and to recent tokens, with only a minority of heads requiring global attention (Tang et al., 2025; Xiao et al., 2025). We draw two insights from this line of work for the design of hybrid attention. First, the window that SWA heads actually need may be far shorter than the windows used in existing architectures (Fu et al., 2025; Wang et al., 2025). How short is an empirical question that has not been answered systematically. Second, a head needs some way to control how much it contributes to the attention output. Pretrained LLMs often obtain this control implicitly by attending to sink tokens (Bondarenko et al., 2023; Barbero et al., 2025; Sun et al., 2026). An SWA head loses access to this mechanism once the sink token falls outside its window, and the problem becomes more acute as the window size decreases. This suggests that short windows require for an explicit gating mechanism.

Guided by these insights, we study *Gated Hybrid Attention* (GHA), which combines very short sliding windows with global attention and applies input-dependent output gates to both head groups.

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



(a) Full Global Attention



(b) Gated Hybrid Attention

Model	Gated	50% SWA Heads $w = 32$	50% SWA Heads $w = 1024$	PPL ↓	Avg. Accuracy ↑
Llama 3 8B	(Baseline)			10.42	58.3
	✓			10.09 (−0.33)	58.3 (0.0)
	✓	✓		**9.89** (−0.53)	**59.4** (+1.2)
	✓		✓	10.06 (−0.36)	58.8 (+0.5)
Qwen3 4B	(Baseline)			10.52	56.8
	✓			10.40 (−0.12)	56.6 (−0.2)
	✓	✓		**10.09** (−0.43)	**58.0** (+1.2)
	✓		✓	10.24 (−0.28)	56.8 (0.0)

(c) Training perplexity and downstream accuracy

Figure 1: Gated Hybrid Attention (GHA): design and larger-scale results. (a) Full global attention: every head can attend to all visible positions. (b) GHA: a subset of heads retain global attention, while the remaining heads attend only to the w most recent tokens. Input-dependent output gates (G) regulate both groups. (c) Training perplexity (PPL) and average downstream accuracy (Avg. Accuracy) after 200,000 steps (approximately 400B tokens). In both model families, GHA with a 32-token window achieves the lowest PPL and highest average accuracy among the configurations shown. Avg. Accuracy is the mean zero-shot accuracy over 21 metrics from 13 benchmarks.

Figure 1(a,b) contrasts GHA with the global-attention baseline. We train 1B-parameter Llama-3-style models on 100B tokens, sweeping the window size, the fraction of SWA heads, and the presence of gating. Our main findings are:

- **Short windows beat long windows.** Across nearly all tested hybrid configurations, a 32-token window outperforms both the global-attention baseline and a 1024-token window. The only exception is the extreme case where just 1/8 layers retain global attention, where the 1024-token window is slightly better.
- **Gating helps short windows most.** Conditional gating lowers perplexity in most settings, with the largest gains at short windows, consistent with the proposed interaction between window size and output control.

Both findings hold across ablations of head configurations, architectural variants, sequence lengths, and training budgets. We extend the comparison to Llama 3 8B (Grattafiori et al., 2024) and Qwen3 4B (Yang et al., 2025) variants trained on approximately 400B tokens each. As Figure 1(c) shows, GHA with a 32-token window achieves the lowest training perplexity and the highest average downstream accuracy among the tested configurations in both model families, outperforming both global attention and the gated 1024-token hybrid.

2 Attention Sinks and StreamingLLM

In this section, we first review the standard formulation of multi-head attention, which fixes the notation used throughout the paper. We then draw two insights from recent work on attention sinks and StreamingLLM. Each insight motivates one component of GHA: gated attention and sliding window attention.

2.1 Attention

Consider a sequence of T tokens, and let $\mathbf{z}_1, \ldots, \mathbf{z}_\mathrm{T} \in \mathbb{R}^{\mathrm{d_{model}}}$ denote their representations at the input to multi-head attention. Stacking these vectors as rows gives the input matrix $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_\mathrm{T}]^\top \in \mathbb{R}^{\mathrm{T} \times \mathrm{d_{model}}}$. Multi-head attention consists of $\mathrm{N_{head}}$ heads, each of dimension $\mathrm{d_{head}}$. For head $h \in \{1, \ldots, \mathrm{N_{head}}\}$, the input is projected into queries, keys, and values using learned matrices $\mathbf{W}_\mathrm{Q}^{(h)}, \mathbf{W}_\mathrm{K}^{(h)}, \mathbf{W}_\mathrm{V}^{(h)} \in \mathbb{R}^{\mathrm{d_{model}} \times \mathrm{d_{head}}}$:

$$\mathbf{Q}^{(h)} = \mathbf{Z}\mathbf{W}_\mathrm{Q}^{(h)}, \qquad \mathbf{K}^{(h)} = \mathbf{Z}\mathbf{W}_\mathrm{K}^{(h)}, \qquad \mathbf{V}^{(h)} = \mathbf{Z}\mathbf{W}_\mathrm{V}^{(h)}. \tag{1}$$

The attention weights and head output are then computed as

$$\mathbf{A}^{(h)} = \mathrm{softmax}\left(\frac{\mathbf{Q}^{(h)}\mathbf{K}^{(h)\top}}{\sqrt{\mathrm{d}_{\mathrm{head}}}} + \mathbf{M}\right), \tag{2}$$

$$\mathbf{O}^{(h)} = \mathbf{A}^{(h)}\mathbf{V}^{(h)}, \tag{3}$$

where $\mathrm{softmax}$ is applied row-wise, so each row of $\mathbf{A}^{(h)}$ is a valid probability distribution. The additive mask $\mathbf{M} \in \{0, -\infty\}^{\mathrm{T}\times\mathrm{T}}$ specifies which positions each query can attend to. In language models, the mask is causal,

$$\mathbf{M}_{tj} = \begin{cases} 0 & \text{if } j \le t, \\ -\infty & \text{otherwise}, \end{cases} \tag{4}$$

preventing attention to future tokens. For clarity, we omit positional encoding from these equations.

Each head output is projected back to the model dimension using $\mathbf{W}_{\mathrm{O}}^{(h)} \in \mathbb{R}^{\mathrm{d}_{\mathrm{head}}\times\mathrm{d}_{\mathrm{model}}}$. Summing these projected outputs gives the output of multi-head attention:

$$\mathcal{F}_{\mathrm{attention}}(\mathbf{Z}) = \sum_{h=1}^{\mathrm{N}_{\mathrm{head}}} \mathbf{O}^{(h)}\mathbf{W}_{\mathrm{O}}^{(h)}. \tag{5}$$

2.2 Insight 1: Sinks Act as Implicit Gates

In pretrained LLMs, many heads assign large attention weights to a few tokens regardless of their semantic relevance. These tokens act as *attention sinks* (Xiao et al., 2024; Gu et al., 2025). A head must distribute its softmax mass even when no available content is useful, and prior work interprets attention to sinks as a way to make small updates to the residual stream (Bondarenko et al., 2023; Gu et al., 2025; Barbero et al., 2025). Sun et al. (2026) further show that sink tokens carry near-constant hidden representations that persist across layers and are largely independent of the input. Both the key and the value of a sink are therefore fixed: the key acts as a constant bias against which each query is scored, and the value is a constant vector that the head can mix into its output.

To see the effect, let $\mathbf{o}_t^{(h)}$ and $\mathbf{v}_j^{(h)}$ denote row t of $\mathbf{O}^{(h)}$ and row j of $\mathbf{V}^{(h)}$, and let $a_{tj}^{(h)} = \mathbf{A}_{tj}^{(h)}$. For a sink at position $s < t$, the head output decomposes as

$$\mathbf{o}_t^{(h)} = a_{ts}^{(h)}\mathbf{v}_s^{(h)} + \big(1 - a_{ts}^{(h)}\big)\tilde{\mathbf{o}}_t^{(h)}, \qquad \tilde{\mathbf{o}}_t^{(h)} = \sum_{\substack{j\le t \\ j\neq s}} \frac{a_{tj}^{(h)}}{1 - a_{ts}^{(h)}}\mathbf{v}_j^{(h)}, \tag{6}$$

where $\tilde{\mathbf{o}}_t^{(h)}$ is the renormalized output of the non-sink tokens. Since $\mathbf{v}_s^{(h)}$ is fixed, the only input-dependent quantity on the sink side is $a_{ts}^{(h)}$, and $1 - a_{ts}^{(h)}$ acts as a scalar gate on the content the head reads from the remaining tokens. In other words, the model learns to gate each head's contribution, but does so indirectly, through softmax mass spent on a token that carries no content. Rather than rely on this implicit mechanism, we make the gate explicit.

Gated attention. We adopt gated attention, which adds an input-dependent output gate to every head (Qiu et al., 2025). Alongside $\mathbf{W}_{\mathrm{Q}}^{(h)}, \mathbf{W}_{\mathrm{K}}^{(h)}, \mathbf{W}_{\mathrm{V}}^{(h)}$, each head has a fourth learned matrix $\mathbf{W}_{\mathrm{G}}^{(h)} \in \mathbb{R}^{\mathrm{d}_{\mathrm{model}}\times\mathrm{d}_{\mathrm{head}}}$, and the head output in Equation 3 is replaced by

$$\mathbf{O}^{(h)} = \sigma\Big(\mathbf{Z}\mathbf{W}_{\mathrm{G}}^{(h)}\Big) \odot \mathbf{A}^{(h)}\mathbf{V}^{(h)}, \tag{7}$$

where σ is the elementwise sigmoid and $\odot$ the Hadamard product; $\mathbf{A}^{(h)}$ and the output projection $\mathbf{W}_{\mathrm{O}}^{(h)}$ are unchanged. The gate plays the same role as $1 - a_{ts}^{(h)}$ in Equation 6: it scales how much of the attended content enters the residual stream, as a function of the current token. Prior work shows that such gates improve performance and reduce attention sinks in full-attention models (Bondarenko et al., 2023; Qiu et al., 2025). Unlike the implicit gate, however, the explicit gate is per-dimension rather than per-head, and it does not require any sink token to be present in the context. The last property is what matters for our design, as we explain next.

2.3 Insight 2: Models Learn to Use Small Windows

StreamingLLM shows that keeping only the initial sinks plus a sliding window of recent tokens preserves stable language modeling, whereas evicting the sinks causes severe degradation (Xiao et al., 2024). Under Insight 1, this is expected: the sinks are needed to control the head outputs. DuoAttention and RazorAttention further find that most heads rely only on recent tokens and sinks, while a small subset of retrieval heads need global access (Xiao et al., 2025; Tang et al., 2025), and Sun et al. (2026) show that sinks bias individual heads toward short-range dependencies. In other words, pretrained models already use short effective windows for most heads, even though they were trained with full attention. This suggests building short windows in from the start; and since explicit gates take over the control role of the sinks, the windowed heads no longer need to attend to sink tokens at all.

Sliding window attention. We adopt sliding window attention for a subset of heads. It restricts each query to the w most recent positions by tightening the causal mask in Equation 4 to

$$\mathbf{M}_{tj} = \begin{cases} 0 & \text{if } t - \mathrm{w} < j \leq t, \\ -\infty & \text{otherwise,} \end{cases} \tag{8}$$

with Equation 3 otherwise unchanged. The cost per query is then $\mathcal{O}(\mathrm{w})$ instead of $\mathcal{O}(\mathrm{T})$, and the KV cache of the head is bounded by w entries. Following the split observed in DuoAttention (Xiao et al., 2025), we designate a fraction of heads in each layer as *SWA heads* that use the mask in Equation 8, while the remaining *global heads* keep the causal mask in Equation 4 so that retrieval over the full context remains possible. Two choices distinguish our use from prior work. First, because explicit gates replace the control role of the sinks (Insight 1), the SWA heads do not retain any initial tokens. Second, we apply the window during pretraining rather than as a post-hoc approximation of a full-attention model. Whether such short windows suffice, and how the answer depends on the fraction of SWA heads, are the empirical questions we study in the following sections.

3 Experiments

In this section, we test the two components introduced in the previous section. Our base model is a 1B-parameter Llama-3-style model (Grattafiori et al., 2024) with 16 layers, hidden size 2048, and 32 query heads sharing 8 key-value heads of dimension 64. We train on DataComp-LM (Li et al., 2024) for 50,000 steps at sequence length 8192 with 2M tokens per step, roughly 100B tokens in total, and report the final held-out validation perplexity. Full global attention with and without gating serves as the two baselines. Implementation details are given in Appendix A.

Gating and short windows help each other. In the first set of experiments, we replace half of the attention heads with SWA under two layouts: a *head-level* layout, in which half of the heads in every layer use SWA, and a *layer-level* layout, in which layers alternate between SWA and global attention. For each layout, we sweep the window size w from 2 to 4096 with and without gating; a window of 8192 would recover full global attention. Figure 2 shows the results. Under both layouts, perplexity is lowest at window sizes between 16 and 32, and gating helps short windows far more than long ones. This is consistent with Insight 1: once the initial tokens fall outside the window, the explicit gate takes over the control that the sinks used to provide. We use $\mathrm{w} = 32$ as the default in what follows.

More than half of the heads can use SWA at no cost. In the second set of experiments, we vary the fraction of SWA heads with gating enabled, fixing $\mathrm{w} = 32$ and using $\mathrm{w} = 1024$ as a reference. We again consider the head-level and layer-level layouts. The left and center panels of Figure 3 show the results. Across all settings, $\mathrm{w} = 32$ outperforms $\mathrm{w} = 1024$, except in the extreme case where only two global attention layers remain. With $\mathrm{w} = 32$, perplexity is best at an SWA share of around 50%, and raising the share to 70% degrades it only slightly. We adopt a uniform 50% head-level split as a simple default in what follows.

SWA is more effective in early layers. In the third set of experiments, we hold the total SWA share fixed at 50% and redistribute it across depth. The right panel of Figure 3 shows that concentrating SWA heads in early layers improves perplexity, whereas concentrating them in later layers

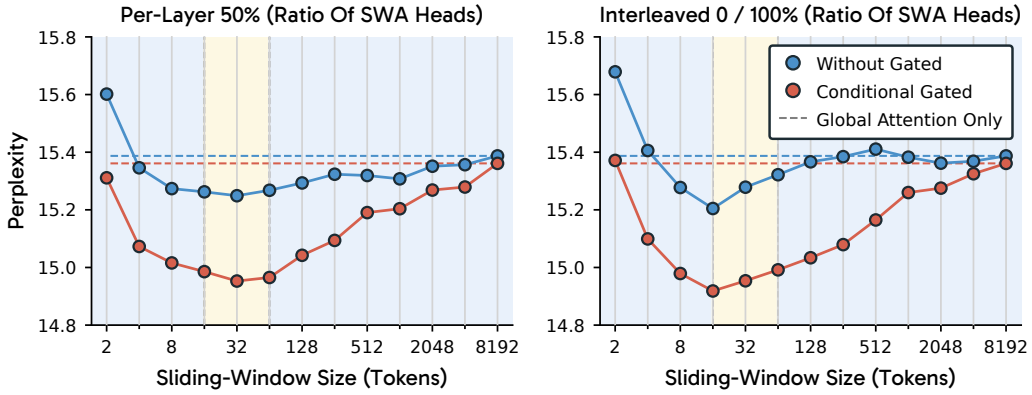


Figure 2: Validation perplexity across window sizes, with and without gating. *(Left)* Head-level layout: half of the heads in every layer use SWA. *(Right)* Layer-level layout: layers alternate between SWA and global attention. Dashed lines show full global attention with the corresponding gate setting.

is markedly worse, suggesting that global access matters most in later layers. Since the gain over the uniform split is small, however, we keep the uniform split for simplicity in the remaining experiments.

3.1 Configuration Notation

We refer to the resulting architecture as Gated Hybrid Attention and write $\mathrm{GHA}(\mathrm{w}, p)$ for the configuration with window size w and a uniform fraction p of SWA heads in every layer, with gating on both head groups. For example, $\mathrm{GHA}(32, 50\%)$ assigns half of the heads in each layer to a 32-token sliding window and keeps the other half global. This is the default configuration selected above, and we use it in the ablations that follow unless stated otherwise.

3.2 Larger Models and Longer Training

Finally, we test whether the findings transfer to larger models trained for longer. We train Llama 3 8B and Qwen3 4B variants (Grattafiori et al., 2024; Yang et al., 2025) from scratch for 200,000 steps,

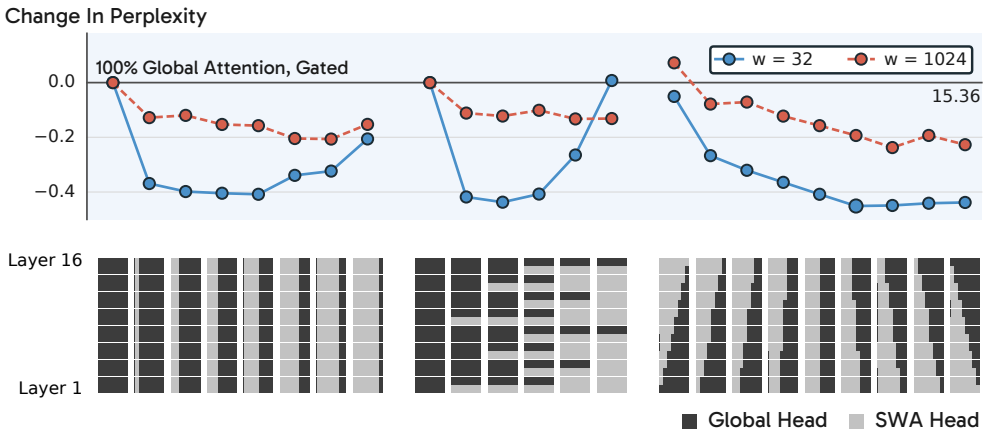


Figure 3: Effect of the fraction and placement of SWA heads at $\mathrm{w} = 32$ (solid) and $\mathrm{w} = 1024$ (dashed), with gating enabled. Columns show the head allocation across the 16 layers, from input (bottom) to output (top); dark denotes global heads and light denotes SWA heads. *(Left)* Varying the fraction of SWA heads within each layer. *(Center)* Varying the fraction of SWA layers. *(Right)* Redistributing a fixed 50% share across depth.

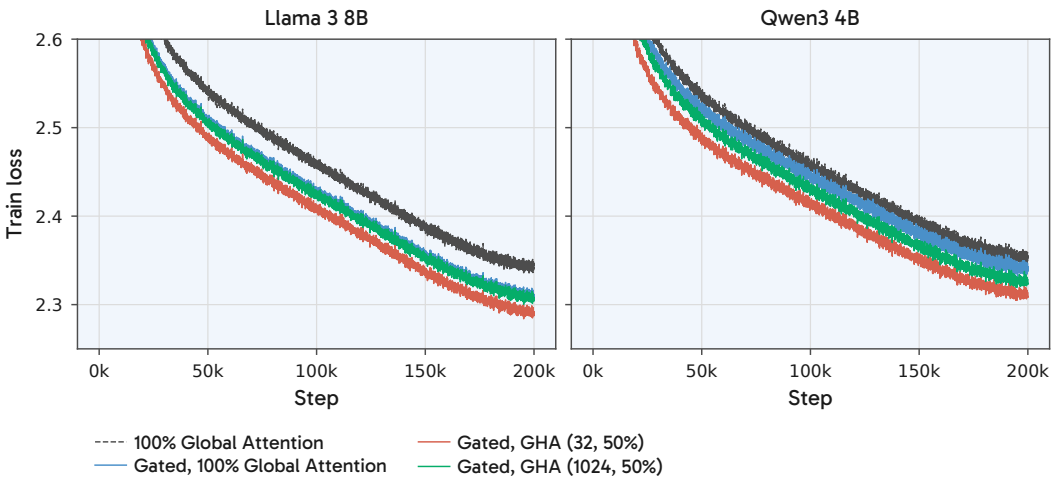


Figure 4: Training loss over 200,000 steps (approximately 400B tokens) for Llama 3 8B (*(Left)*) and Qwen3 4B (*(Right)*), comparing full global attention with and without gating, gated $\mathrm{GHA}(32, 50\%)$, and gated $\mathrm{GHA}(1024, 50\%)$. In both families, $\mathrm{GHA}(32, 50\%)$ has the lowest loss throughout training, and the gap to global attention opens early and persists.

roughly 400B tokens, four times the budget of the base runs. For each family we compare full global attention with and without gating, $\mathrm{GHA}(32, 50\%)$, and $\mathrm{GHA}(1024, 50\%)$. The Qwen3 variant additionally uses QK normalization and a 32,768-token context, so it also tests the two components under an architecture and sequence length different from the base setup. Implementation details are given in Appendix A.

Figure 4 shows the training loss. In both families, the ordering is established early and never changes: $\mathrm{GHA}(32, 50\%)$ has the lowest loss throughout, followed by $\mathrm{GHA}(1024, 50\%)$, gated global attention, and ungated global attention. The gap between $\mathrm{GHA}(32, 50\%)$ and global attention opens within the first 50,000 steps and does not close with further training, so the advantage of short windows is not an artifact of a small training budget. The final numbers in Figure 1(c) confirm this: $\mathrm{GHA}(32, 50\%)$ lowers training perplexity by 0.53 on Llama 3 8B and 0.43 on Qwen3 4B relative to global attention, compared with 0.36 and 0.28 for $\mathrm{GHA}(1024, 50\%)$, and raises average zero-shot accuracy over 21 metrics by 1.2 points in both families. Gated global attention, by contrast, improves perplexity but not accuracy, and $\mathrm{GHA}(1024, 50\%)$ gains only 0.5 points on Llama 3 8B and nothing on Qwen3 4B. Short windows and gating together, rather than either alone, are what carry over to downstream performance at this scale.

4 Ablations

In this section, we test how robust the findings of Section 3 are to changes in the gating mechanism, the head configuration, and other architectural choices. Unless specified otherwise, experiments use the base model and training setup of Section 3 with the default $\mathrm{GHA}(32, 50\%)$ configuration. We also report $\mathrm{GHA}(1024, 50\%)$ throughout to separate effects specific to short windows from effects of windowing in general.

4.1 Gating Mechanism

Both head groups need to be gated. Table 1 reports the effect of gating each head group separately. At $\mathrm{w} = 32$, gating only the SWA heads or only the global heads falls well short of gating both. At $\mathrm{w} = 1024$, gating both groups still helps, but the total gain is only 0.11, compared with 0.30 at $\mathrm{w} = 32$. Gating therefore interacts most strongly with short windows, as Insight 1 predicts: it is the SWA heads, whose context no longer contains the initial sinks, that gain the most from an explicit gate.

The gate must depend on the input. We replace the input-dependent gate with a learned but input-independent gate, a static sigmoid vector applied to every token, and repeat the window-size sweep. Figure 5 shows the results. The static gate performs almost identically to no gate at all across

Table 1: Effect of gating each head group in $\mathrm{GHA}(\mathrm{w}, 50\%)$. Rows specify gating on global heads; columns specify gating on SWA heads. Lower validation perplexity is better.

Global Heads	SWA Heads			
	$\mathrm{w} = 32$		$\mathrm{w} = 1024$	
	Not Gated	Gated	Not Gated	Gated
Not Gated	15.25	15.09	15.31	15.23
Gated	15.14	**14.95**	15.23	15.20

all window sizes, well behind the input-dependent gate. A static per-dimension rescaling therefore does not reproduce the benefit; the gain comes from conditioning the gate on the current token, just as the sink mechanism does through $a_{ts}^{(h)}$.

4.2 Head Configuration

In this subsection, we vary the configuration of one head group at a time, while the other group retains the default of 16 query heads, 4 key-value heads, and head dimension 64.

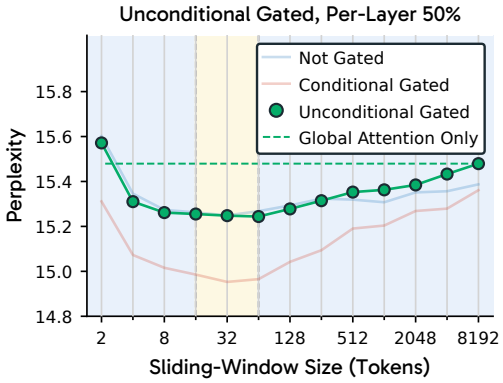


Figure 5: Perplexity across window sizes for $\mathrm{GHA}(\mathrm{w}, 50\%)$ with no gate, unconditional gate, and conditional gate.

Both head groups benefit from more key-value heads. In the first set of experiments, we vary the number of key-value heads in one group while keeping its number of query heads fixed, which changes the group size of grouped-query attention. The left panel of Figure 6 shows the results. Both head groups benefit almost equally from additional key-value heads, so grouped-query attention does not favor either group.

Head width matters more for global heads. In the second set of experiments, we vary the head dimension of one group at fixed total width, so that fewer, wider heads trade off against more, narrower heads. The center panel of Figure 6 shows the results. Reducing the head dimension of the global heads substantially worsens perplexity, whereas reducing it for the SWA heads at $\mathrm{w} = 32$ has almost no effect. At $\mathrm{w} = 1024$, the SWA heads also benefit from wider heads, suggesting that head width matters in proportion to how much context a head must summarize.

SWA heads need value width more than query-key width. Since head width matters, we next vary the query-key dimension and the value dimension of one group independently. Figure 7 shows the results. Global heads benefit roughly equally from increasing either dimension. SWA heads, in contrast, benefit much more from a larger value dimension than from a larger query-key dimension, at both $\mathrm{w} = 32$ and $\mathrm{w} = 1024$. With only a few dozen keys to compare, an SWA head has little use for extra query-key capacity, but still needs value capacity to carry content into the residual stream.

Short windows still need positional encoding, but are less sensitive to it. In the last set of experiments, we vary the RoPE setting of each group. The right panel of Figure 6 shows the results. Removing RoPE from either group worsens perplexity at both window sizes, so even a 32-token window benefits from positional information. Reducing the RoPE base from 5×10^5 to 5×10^3, however, hurts the global heads considerably more than the SWA heads. Global heads are thus much more sensitive to the positional encoding than SWA heads.

4.3 Other Architectural Changes

Table 2 compares full global attention, GHA(32, 50%), and GHA(1024, 50%), each with and without gating, under three common architectural changes: QK normalization (Henry et al., 2020; Dehghani et al., 2023), sandwich normalization (Ding et al., 2021), and a short causal convolution on queries, keys, and values (Fu et al., 2023; Gu & Dao, 2023). QK normalization improves every

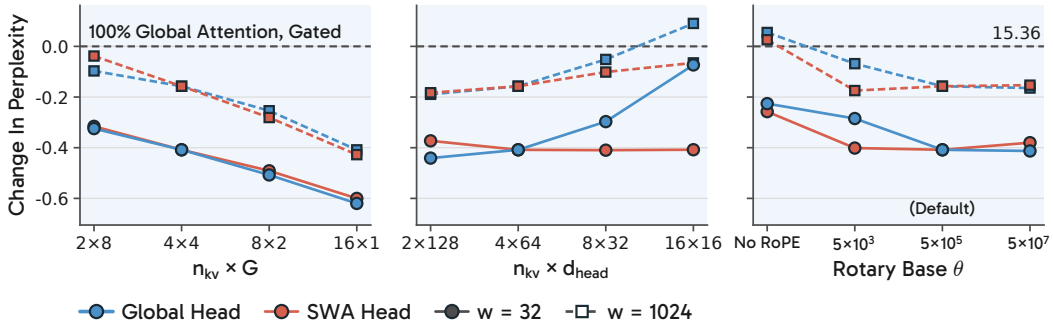


Figure 6: Validation perplexity when varying the configuration of one head group (SWA at $w = 32$ or $w = 1024$, or global) while the other group keeps the default configuration. *(Left)* Number of key-value heads. *(Center)* Head dimension at fixed total width. *(Right)* Different RoPE setting.

model, full attention most; the advantage of gated $\mathrm{GHA}(32, 50\%)$ over gated full attention shrinks from 0.41 to 0.18 but remains. Sandwich normalization, which adds RMS normalization to the attention and feed-forward outputs before their residual additions, improves full attention but worsens every GHA variant. A width-4 causal depthwise convolution with SiLU on queries, keys, and values worsens all six models, with gated $\mathrm{GHA}(32, 50\%)$ the least affected ($+0.07$ versus $+0.20$ for gated full attention), suggesting that its SWA heads already capture the short-range structure a convolution would provide.

5 Related Work

Dual attention patterns Transformers (Vaswani et al., 2017) place no structural prior on where information flows, learning long-range dependencies at quadratic cost. Yet analysis of trained models repeatedly finds attention to be highly non-uniform: many heads concentrate on nearby tokens (Clark et al., 2019b; Voita et al., 2019), a few positions absorb disproportionate mass as attention sinks (Xiao et al., 2024), and pruning distant connections often leaves quality largely intact (Child et al., 2019; Beltagy et al., 2020). Convolutional and recurrent architectures encode locality by construction, and state-space models (Gu & Dao, 2023) reach strong language modeling performance with an explicit local-plus-recurrent bias. If trained attention gravitates toward locality anyway, a natural question is whether locality should be built into the architecture.

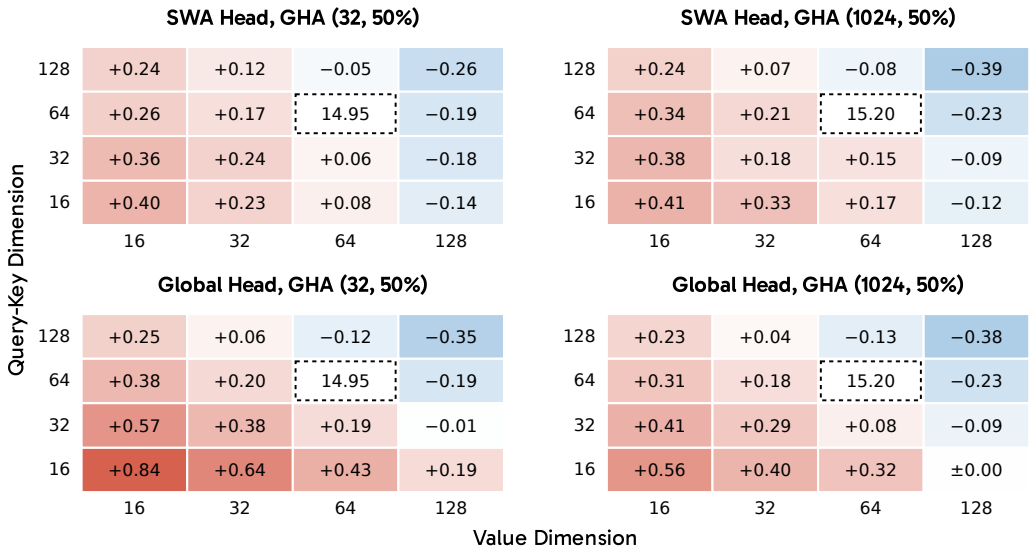


Figure 7: Effect of query–key and value dimensions. Cells report changes in validation perplexity relative to the default $d_{qk} = d_v = 64$ configuration within each panel. The outlined reference cell reports its absolute perplexity. Top: varying SWA heads; bottom: varying global heads. Left: $w = 32$; right: $w = 1024$. The other head group remains at its default configuration.

Table 2: Effect of three architectural changes on full global attention and on $\mathrm{GHA}(\mathrm{w}, 50\%)$. QK-norm helps every model, full attention most; sandwich norm helps full attention but hurts every GHA variant; a width-4 causal convolution on queries, keys and values hurts every model. Validation perplexity after 50,000 steps at sequence length 8192; the small figure in parentheses after each value is the change from the baseline row. Bold marks the best value in each row.

	Global Attention		50% SWA Heads			
			$w = 32$		$w = 1024$	
	Not Gated	Gated	Not Gated	Gated	Not Gated	Gated
Baseline	15.39	15.36	15.25	**14.95**	15.31	15.20
QK-Norm	15.15 (−0.23)	15.06 (−0.30)	15.12 (−0.13)	**14.88** (−0.07)	15.31 (+0.00)	15.07 (−0.14)
Sandwich Norm	15.38 (−0.01)	15.22 (−0.14)	15.39 (+0.14)	**15.15** (+0.19)	15.49 (+0.18)	15.27 (+0.06)
Short Convolution	15.94 (+0.56)	15.56 (+0.20)	15.29 (+0.04)	**15.02** (+0.07)	15.72 (+0.41)	15.37 (+0.17)

Sliding window attention Sliding window attention restricts each query to the w most recent positions, reducing the cost of self-attention from quadratic to $O(nw)$. Sparse Transformers established the effectiveness of local patterns (Child et al., 2019); Longformer (Beltagy et al., 2020) paired sliding windows with a few global tokens, and BigBird (Zaheer et al., 2020) added random connections with universality guarantees for sparse attention. The design is now used at production scale: Mistral (Jiang et al., 2023) applies it in every layer, while Gemma 2 (Gemma Team et al., 2024) and Gemma 3 (Gemma Team, 2025) interleave local and global layers at fixed ratios. Persistent attention sinks extend local windows to unbounded context (Xiao et al., 2024), and Swin (Liu et al., 2021) popularized shifted local windows in vision. These results suggest locality is a useful architectural prior rather than merely a compute-saving approximation, motivating a closer look at which layers and heads actually benefit from global receptive fields.

Gated attention Gating has long served to modulate information flow, from LSTMs (Hochreiter & Schmidhuber, 1997) to highway networks (Srivastava et al., 2015). The Gated Attention Unit (Hua et al., 2022) multiplies attention outputs by a content-dependent gate, and linear and recurrent variants rely on gating to recover expressivity lost without softmax: Gated Linear Attention (Yang et al., 2023) uses data-dependent forget gates, RetNet (Sun et al., 2023) exponential decay gates, and Mamba (Gu & Dao, 2023) input-dependent selective gates. Gating has recently returned to softmax attention itself: the Forgetting Transformer (Lin et al., 2025) adds a learned forget gate to the attention matrix, and Qiu et al. (2025) show that a sigmoid gate on attention outputs improves quality, length generalization, and attention-sink behavior in large language models. In each case, gating is a lightweight learned routing mechanism, orthogonal to the choice of receptive field or kernel.

6 Conclusion

We revisited the assumption that hybrid attention trades quality for efficiency and that larger sliding windows are therefore always better. Two observations about pretrained LLMs suggested otherwise: most heads use only a short effective window, and the attention sinks they rely on act as an implicit output gate rather than a source of content. This led to Gated Hybrid Attention, which pairs very short sliding windows with global attention and replaces the sink mechanism with an explicit, input-dependent gate on every head.

Across 1B-parameter models trained on 100B tokens, a 32-token window outperforms both full global attention and a 1024-token window, and gating helps most on the short-window heads that have lost their sinks. These findings survive ablations over head configuration, normalization, positional encoding, and training budget, and they carry over to Llama 3 8B and Qwen3 4B variants trained on roughly 400B tokens, where $\mathrm{GHA}(32, 50\%)$ achieves the lowest perplexity and a 1.2-point gain in average downstream accuracy over global attention.

Two questions remain open. Our evaluation centers on perplexity and short-context benchmarks, so whether the advantage persists on long-context retrieval deserves a dedicated study. And while our results indicate that a short window is a useful inductive bias rather than a cheap approximation, they do not yet explain why; understanding what short-window heads learn that global heads do not is a natural next step.

AI USE STATEMENT

We used generative AI tools, including Codex and Claude, to assist with interpreting experimental results and articulating the conceptual relationship between attention sinks, conditional gating, and short attention windows. We also used these tools to identify and summarize relevant literature, draft and revise manuscript text, refine mathematical notation, format references and figures in LaTeX, and check consistency between the manuscript and the supplied code, configurations, and results. We did not use generative AI to design experiments, implement training or evaluation code, or prepare datasets. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *EMNLP*, pp. 4895–4901, 2023. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

Federico Barbero, Álvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do LLMs attend to the first token? *arXiv preprint arXiv:2504.02732*, 2025. URL https://arxiv.org/abs/2504.02732.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. URL https://arxiv.org/abs/2004.05150.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. In *NeurIPS*, volume 36, pp. 75067–75096, 2023. URL https://arxiv.org/abs/2306.12929.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. URL https://arxiv.org/abs/1904.10509.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019a.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *ACL*, 2019b.

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

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, 2005.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024. URL https://arxiv.org/abs/2307.08691.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *ICML*, 2023. URL https://arxiv.org/abs/2302.05442.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. CogView: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021. URL https://arxiv.org/abs/2105.13290.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels. In *MLSys*, 2025. URL https://arxiv.org/abs/2412.05496.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. In *ICLR*, 2023. URL https://arxiv.org/abs/2212.14052.

Tianyu Fu, Haofeng Huang, Xuefei Ning, Genghan Zhang, Boju Chen, Tianqi Wu, Hongyi Wang, Zixiao Huang, Shiyao Li, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Mixture of attention spans: Optimizing LLM inference efficiency with heterogeneous sliding-window lengths. In *COLM*, 2025. URL https://arxiv.org/abs/2406.14909.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

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

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

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *ICLR*, 2025. URL https://arxiv.org/abs/2410.10781.

Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of EMNLP*, 2020. URL https://arxiv.org/abs/2010.04245.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *ICML*, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023. doi: 10.1145/3600006.3613165. URL https://arxiv.org/abs/2309.06180.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. DataComp-LM: In search of the next generation of training sets for language models. In *NeurIPS*, volume 37, 2024. URL https://arxiv.org/abs/2406.11794.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *ACL*, 2022.

Zhixuan Lin, Evgenii Nikishin, Xu Owen He, and Aaron Courville. Forgetting transformer: Softmax attention with a forget gate. *arXiv preprint arXiv:2503.02130*, 2025.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

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

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

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. In *ACL*, pp. 4885–4901, 2020.

Olmo Team, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025. URL https://arxiv.org/abs/2512.13961.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *NAACL*, 2019.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025. URL https://arxiv.org/abs/2505.06708.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 2021.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. URL https://arxiv.org/abs/2002.05202.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Shangwen Sun, Alfredo Canziani, Yann LeCun, and Jiachen Zhu. The spike, the sparse and the sink: Anatomy of massive activations and attention sinks. *arXiv preprint arXiv:2603.05498*, 2026. URL https://arxiv.org/abs/2603.05498.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Hanlin Tang, Yang Lin, Jing Lin, Qingsen Han, Danning Ke, Shikuan Hong, Yiwu Yao, and Gongyi Wang. RazorAttention: Efficient KV cache compression through retrieval heads. In *ICLR*, 2025. URL https://openreview.net/forum?id=tkiZQlL04w.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. URL https://arxiv.org/abs/1706.03762.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *ACL*, 2019.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *NeurIPS*, 2019.

Bailin Wang, Chang Lan, Chong Wang, and Ruoming Pang. RAttention: Towards the minimal sliding window size in local-global attention models. *arXiv preprint arXiv:2506.15545*, 2025. URL https://arxiv.org/abs/2506.15545.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024. URL https://arxiv.org/abs/2309.17453.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *ICLR*, 2025. URL https://openreview.net/forum?id=cFu7ze7xUm.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *ACL*, pp. 4791–4800, 2019.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *NeurIPS*, volume 32, 2019. URL https://arxiv.org/abs/1910.07467.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and BV Durme. Record: Bridging the gap between, human and machine commonsense reading comprehension, corr abs/1810.12885, 2018. *arXiv preprint arXiv:1810.12885*, 1810.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. PyTorch FSDP: Experiences on scaling fully sharded data parallel. *Proceedings of the VLDB Endowment*, 16(12), 2023. URL https://arxiv.org/abs/2304.11277.

A Implementation Details

Model architecture. Our base model is a decoder-only Transformer following the Llama design (Grattafiori et al., 2024). It uses pre-attention and pre-feed-forward RMS normalization (Zhang & Sennrich, 2019), a SwiGLU feed-forward block (Shazeer, 2020), and grouped-query attention (Ainslie et al., 2023) with four query heads per key-value head. Rotary position embeddings (Su et al., 2024) are applied to the full query and key dimensions, with the same RoPE base for global and SWA heads. The input embedding and language-model output projection are untied in the Llama variants. All linear projections are bias-free, and no dropout is used. Table 3 summarizes the architectures and training budgets.

Attention layouts and gates. In $\mathrm{GHA}(32, 50\%)$, each layer contains a global group and an SWA group, each with 16 query heads and four key-value heads. The two groups have separate query, key, value, and output projections; their projected outputs are summed before the residual addition. Conditional gates are computed from the normalized input to attention and applied elementwise to each head's output before its output projection (Qiu et al., 2025). Gating is enabled for both groups unless an ablation specifies otherwise. The SWA window includes the current token, and no initial tokens are retained outside the window. In the alternating-layer layout, odd-numbered layers use SWA and even-numbered layers use global attention, with 32 query heads and eight key-value heads in each layer. Changing the window size alone leaves the projection dimensions unchanged; adding conditional gates introduces additional learned parameters.

Table 3: Model configurations and training budgets. Head counts are totals per layer before splitting into global and SWA groups. The base configuration is used for the main sweeps and ablations unless otherwise stated.

Setting	Base Llama	Llama 3 8B	Qwen3 4B
Decoder layers	16	32	36
Hidden size	2048	4096	2560
Feed-forward width	8192	14,336	9728
Query / key-value heads	32 / 8	32 / 8	32 / 8
Query-key / value head dimension	64 / 64	128 / 128	128 / 128
Vocabulary size	128,256	128,256	151,936
RoPE base	5×10^5	5×10^5	10^6
RMSNorm epsilon	10^{-5}	10^{-5}	10^{-6}
Query-key normalization	No	No	Yes
Tied input / output embeddings	No	No	Yes
Sequence length	8192	8192	32,768
Sequences per optimizer step	256	256	64
Optimizer steps	50,000	200,000	200,000
Training tokens (billions)	104.86	419.43	419.43

Table 4: Optimization settings shared by the base and larger-scale experiments. The decay schedule spans the training budget of each run.

Setting	Value
Optimizer	AdamW
Peak learning rate	3×10^{-4}
Final learning rate	3×10^{-5}
Warmup steps	2000
Adam coefficients (β_1, β_2)	$(0.9, 0.95)$
Weight decay	0.1
Gradient-norm clipping threshold	1.0
Global batch size (tokens)	2,097,152

Data and training objective. We train from random initialization on DataComp-LM (Li et al., 2024) using next-token cross-entropy. Training examples are read from pretokenized Arrow shards partitioned across distributed workers. For a sequence length of T, the data loader reads $\mathrm{T}+1$ tokens and forms input and target sequences by shifting them by one position. The loss is averaged over tokens and computed in float32. Every optimizer step processes 2,097,152 tokens globally, corresponding to 256 sequences at length 8192 or 64 sequences at length 32,768. The base runs therefore process 104.86B tokens in 50,000 steps, rounded to 100B in the main text.

Optimization. All configurations use AdamW (Loshchilov & Hutter, 2019) with the settings in Table 4. The learning rate increases to 3×10^{-4} over 2000 warmup steps, with multiplier $1-(1-s/2000)^2$ at warmup step s. It then follows a cosine decay to 3×10^{-5} at the end of training. Weight decay is applied to all trainable parameters, including embeddings and normalization weights, and the gradient norm is clipped before each optimizer update. The optimizer settings are shared across attention layouts and gating variants.

Initialization. Query, key, value, conditional-gate, and feed-forward input projections use zero-mean truncated-normal initialization with scale 0.02. At layer ℓ, indexed from one, the attention output and feed-forward output projections use scale $0.02/\sqrt{2\ell}$. Token embeddings and untied language-model output weights use scale $1/\sqrt{\mathrm{d}_{\mathrm{model}}}$, and RMSNorm weights are initialized to one. For the unconditional-gate ablation, each attention output channel has a learned, input-independent gate logit initialized to zero, giving an initial sigmoid gate value of 0.5.

Attention implementation and ablations. Global attention uses FlashAttention (Dao, 2024), while SWA uses PyTorch FlexAttention (Dong et al., 2025) with a cached causal sliding-window block mask. Training uses fully sharded data parallelism (Zhao et al., 2023) with bfloat16 computa-

tion and gradient reduction. The QK-normalization ablation applies RMSNorm separately to queries and keys before RoPE. Sandwich normalization adds RMSNorm to the combined attention output and the feed-forward output before their respective residual additions. The convolution ablation applies width-4 causal depthwise convolutions with SiLU activation to the projected queries, keys, and values before attention. All other settings follow the base configuration unless the experiment explicitly varies them.

Longer training and larger models. The larger-scale comparisons use Llama 3 8B and Qwen3 4B variants (Grattafiori et al., 2024; Yang et al., 2025), each trained for 200,000 steps, or 419.43B tokens. For each family, we compare global attention with and without gating, $\mathrm{GHA}(32, 50\%)$, and $\mathrm{GHA}(1024, 50\%)$. The Qwen3 variant additionally uses query-key RMS normalization before RoPE and ties its input and output embeddings.

Evaluation. The base-model sweeps and ablations report final held-out validation perplexity. For the larger models, we additionally evaluate zero-shot performance on ANLI (Nie et al., 2020) (rounds R1, R2, and R3), ARC-Challenge (Clark et al., 2018), ARC-Easy, BoolQ (Clark et al., 2019a), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), ReCoRD (Zhang et al., 1810), RTE (Dagan et al., 2005; Wang et al., 2019), TruthfulQA (multiple choice) (Lin et al., 2022), WiC (Pilehvar & Camacho-Collados, 2019), and WinoGrande (Sakaguchi et al., 2021). We report accuracy for each ANLI round, BoolQ, RTE, WiC, and WinoGrande; both accuracy and normalized accuracy for ARC-Challenge, ARC-Easy, HellaSwag, OpenBookQA, and PIQA; exact match and F1 for ReCoRD; and MC1 and MC2 scores for TruthfulQA. This gives 21 metrics across 14 task settings, counting the three ANLI rounds separately. The aggregate downstream score is the unweighted mean of these 21 metrics, expressed as a percentage; each reported metric contributes equally, so tasks with multiple metrics receive greater total weight.