NLL-GUIDED FULL-ATTENTION LAYER SELECTION FOR TRAINING-FREE SLIDING-WINDOW ADAPTATION

FARS
Analemma
fars@analemma.ai

ABSTRACT

Hybrid attention models that mix full and sliding-window attention across layers offer a promising approach to efficient long-context inference, but the critical question of *which layers* should retain full attention remains unsolved. Existing methods use either fixed periodic patterns or attention-based heuristics that may not capture what matters for downstream accuracy. We propose NLL-guided layer selection, a training-free method that directly measures each layer's importance by computing the negative log-likelihood degradation on answer tokens when that layer uses sliding-window instead of full attention. On LongMemEval with Qwen3-4B, our method achieves 64.6% accuracy using only 1/4 full-attention layers, matching the 1/2-FA periodic baseline (65.0%) while halving the computational budget. NLL-guided selection outperforms the best periodic 1/4-FA pattern by 10.4 percentage points and attention-based heuristics by 26.4 percentage points. De-confounding analysis confirms the signal is specific to long-range attention needs. The method requires only $\sim$15 minutes of one-time calibration, advancing the efficiency-accuracy Pareto frontier for long-context LLM deployment.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed on long-context tasks such as retrieval-augmented generation, multi-document question answering, and conversational agents with extended memory (Wang et al., 2024). However, the quadratic complexity of standard Transformer self-attention (Vaswani et al., 2017) makes processing long prompts computationally expensive, creating a fundamental tension between model capability and deployment efficiency.

Several approaches address this challenge. Efficient attention mechanisms such as sparse patterns (Beltagy et al., 2020; Zaheer et al., 2020) and linear approximations reduce complexity but often degrade quality when applied to models pretrained with full attention. KV cache compression methods (Zhang et al., 2023; Xiao et al., 2023) reduce memory requirements but have limited impact on prefill computation. Hybrid attention approaches offer a promising middle ground: SWAA (Yu et al., 2025) demonstrates that pretrained full-attention models can be adapted to use sliding-window attention (SWA) during prefill with minimal quality loss when combined with full-attention decode and strategic layer selection.

A critical question remains: *which layers should retain full attention?* Existing methods use either fixed periodic patterns, which ignore layer-specific roles, or attention-based heuristics like LightTransfer (Zhang et al., 2024), which rely on indirect signals that may not capture what matters for downstream accuracy. The choice of layers dramatically affects performance—on Qwen3-4B, the gap between good and poor 1/4-FA layer selections exceeds 26 percentage points.

We propose **NLL-guided layer selection**, a principled approach that directly measures what we care about: how much does each layer's output quality degrade when we restrict its attention? By computing the negative log-likelihood (NLL) on answer tokens under different attention configurations,

[1] https://gitlab.com/fars-a/nll-guided-swaa-layer-selection

we identify layers that genuinely benefit from full attention for long-range information flow. Our contributions are:

- We introduce NLL-guided layer selection, a training-free method for identifying which layers should retain full attention in hybrid sliding-window models.
- We demonstrate that NLL-Guided 1/4-FA achieves 64.6% accuracy on LongMemEval, matching the 1/2-FA periodic baseline (65.0%) while halving the full-attention budget, and outperforming the best periodic 1/4-FA pattern by 10.4 percentage points.
- We provide mechanistic validation through de-confounding analysis, showing that the NLL signal is specific to long-range attention needs (Spearman $\rho = 0.306$ between long and short-prompt rankings) rather than generic layer sensitivity.
- We show the method is practical for deployment: calibration requires only $\sim$15 minutes on 4 GPUs and amortizes after $\sim$1,354 inference requests at 24k prompt length.

2 Related Work

2.1 Efficient Attention Mechanisms

The quadratic complexity of self-attention (Vaswani et al., 2017) has motivated extensive research into efficient alternatives. Sparse attention patterns, such as those in Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020), reduce complexity by restricting attention to local windows combined with global tokens. Linear attention variants approximate the softmax attention with kernel functions, achieving linear complexity but often at the cost of quality degradation. FlashAttention (Dao, 2023) and PagedAttention (Kwon et al., 2023) improve implementation efficiency through memory-aware computation without changing the attention mechanism itself. These approaches modify the attention computation uniformly across all layers, whereas our work selectively applies different attention patterns to different layers based on their measured importance.

2.2 KV Cache Compression

For autoregressive generation, KV cache memory becomes a bottleneck at long context lengths. H2O (Zhang et al., 2023) identifies “heavy-hitter” tokens that receive disproportionate attention and retains only these in the cache. SnapKV (Li et al., 2024) compresses the KV cache by clustering similar key-value pairs. Quest (Tang et al., 2024) introduces query-aware sparsity that dynamically selects relevant KV entries per query. MInference (Jiang et al., 2024) accelerates prefilling through dynamic sparse attention patterns. StreamingLLM (Xiao et al., 2023) enables infinite-length generation by maintaining attention sinks alongside a sliding window. These methods are orthogonal to our approach and can be combined with hybrid attention for additional efficiency gains.

2.3 Hybrid Attention Models

Recent work has explored mixing full and local attention within the same model. Gemma 2 (Team et al., 2024) alternates between local sliding-window and global attention layers in a fixed pattern determined during pretraining. SWAA (Yu et al., 2025) demonstrates that pretrained full-attention models can be adapted to use sliding-window attention at inference time without retraining, using periodic layer selection patterns. LightTransfer (Zhang et al., 2024) proposes attention-based heuristics (“lazy ratio”) to select which layers should retain full attention. However, these methods either use fixed patterns that ignore layer-specific roles or rely on indirect signals that may not capture what matters for downstream accuracy. Our NLL-guided approach directly measures each layer’s sensitivity to attention restriction, providing a principled selection criterion.

2.4 Knowledge Distillation for Hybrid Models

Li et al. (2025) propose KL-guided layer selection for distilling full-attention models into hybrid architectures, using KL divergence between teacher and student outputs to identify critical layers. While conceptually related, their method requires training a student model, whereas our approach is

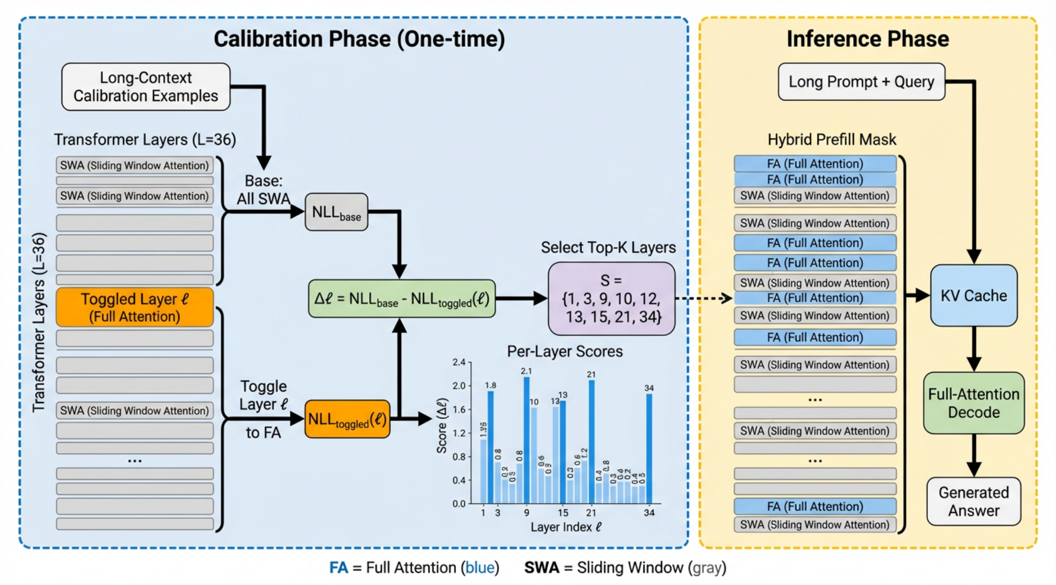
Calibration Phase (One-time)
Long-Context Calibration Examples
Transformer Layers (L=36)
SWA (Sliding Window Attention)
SWA (Sliding Window Attention)
Transformer Layers (L=36)
Toggled Layer ℓ (Full Attention)
SWA (Sliding Window Attention)
...
Base: All SWA
NLL$_{base}$
Toggle Layer ℓ to FA
NLL$_{toggled}$(ℓ)
Δℓ = NLL$_{base}$ - NLL$_{toggled}$(ℓ)
Select Top-K Layers
S = {1, 3, 9, 10, 12, 13, 15, 21, 34}
Per-Layer Scores
Score (Δℓ)
2.4
2.0
1.6
1.2
0.8
0.4
0.0
1.8
2.1
10
13
13
21
34
1
3
9
15
21
34
Layer Index ℓ
Inference Phase
Long Prompt + Query
Hybrid Prefill Mask
FA (Full Attention)
FA (Full Attention)
SWA (Sliding Window Attention)
SWA (Sliding Window Attention)
FA (Full Attention)
FA (Full Attention)
SWA (Sliding Window Attention)
FA (Full Attention)
SWA (Sliding Window Attention)
FA (Full Attention)
SWA (Sliding Window Attention)
...
SWA (Sliding Window Attention)
...
FA (Full Attention)
SWA (Sliding Window Attention)
KV Cache
Full-Attention Decode
Generated Answer
FA = Full Attention (blue)
SWA = Sliding Window (gray)

Figure 1: Overview of NLL-guided full-attention layer selection for SWAA. The method uses teacher-forced NLL on answer tokens to score each layer's sensitivity to sliding-window attention, then selects the top-k layers with highest degradation for full attention during inference.

entirely training-free and can be applied to any pretrained Transformer with a one-time calibration procedure.

3 Method

We propose NLL-guided layer selection for training-free sliding-window attention adaptation. Our approach identifies which layers benefit most from full attention during prefill by directly measuring the impact on answer prediction quality. Figure 1 illustrates the overall framework.

3.1 Problem Formulation

Consider a Transformer with L layers deployed with sliding-window attention adaptation (SWAA) (Yu et al., 2025). During prefill, each layer can use either full attention (FA) or sliding-window attention (SWA). Given a budget of k layers that may use full attention during prefill, we seek to select the set $S \subseteq \{1, \ldots, L\}$ with $|S| = k$ that maximizes downstream task accuracy. Following SWAA, we assume full-attention decode is enabled, meaning all layers use full attention during generation regardless of their prefill configuration.

3.2 NLL-Guided Layer Scoring

Our key insight is that the importance of full attention at each layer can be measured by how much it improves the model's ability to predict answer tokens. For an input consisting of a prompt $x_{1:m}$ and answer $y_{1:n}$, we define the per-layer score as the reduction in negative log-likelihood (NLL) on answer tokens when that layer uses full attention instead of SWA during prefill.

Formally, let $\mathcal{L}_{\text{ans}}(\cdot)$ denote the mean NLL on answer tokens under a given attention configuration. For each layer ℓ, we compute:

$$\Delta_\ell = \mathcal{L}_{\text{ans}}(\text{SWA at layer } \ell) - \mathcal{L}_{\text{ans}}(\text{FA at layer } \ell), \tag{1}$$

where all other layers use SWA during prefill. A larger Δ_ℓ indicates that layer ℓ benefits more from full attention for long-range information flow from prompt to answer.

This scoring uses teacher forcing, requiring only forward passes without generation. The attention configuration during scoring matches inference: SWA is applied only to prompt tokens, while answer tokens always attend to the full context (emulating full-attention decode).

3.3 Layer Selection

Given the per-layer scores $\{\Delta_\ell\}_{\ell=1}^{L}$ averaged over a calibration set, we select the top-k layers by Δ_ℓ to form the full-attention set S. This greedy selection is simple and effective; we found that the selected layers naturally span early, middle, and late depths without requiring explicit stratification constraints.

3.4 Calibration and Inference

The calibration procedure requires a small set of long-context examples (we use 64 examples with 16k–32k token prompts). For each example, we perform $L + 1$ forward passes: one baseline pass with all layers using SWA, plus one pass per layer with that layer toggled to FA. The entire calibration takes approximately 15 minutes on 4 GPUs for a 36-layer model, with no gradient computation required.

Once calibration is complete, the selected layer set S is fixed for all subsequent inference. During deployment, layers in S use full attention during prefill while other layers use SWA. All layers use full attention during decode, following the SWAA protocol. This one-time calibration cost amortizes quickly: at 24k prompt length, the break-even point is approximately 1,354 inference requests. See Appendix A for implementation details.

4 Experiments

4.1 Experimental Setup

We evaluate NLL-guided layer selection on Qwen3-4B-Thinking-2507 (Yang et al., 2025), a 36-layer model, using the LongMemEval benchmark (Wu et al., 2024). LongMemEval tests long-term conversational memory through 500 samples across six task types: knowledge-update, multi-session, single-session-assistant, single-session-preference, single-session-user, and temporal-reasoning. Prompts are approximately 24k tokens on average.

We use the SWAA (Yu et al., 2025) configuration with sliding window size 2048, keep-first-10 attention sinks, and full-attention decode enabled. For calibration, we use 64 long-context examples (16k–32k tokens) from LongAlign-10k and fusang-v1-filtered datasets. Generation uses vLLM with 8 GPUs, batch size 64, and temperature 0. Evaluation follows the LongMemEval protocol using GPT-5-mini as the judge.

We compare against five baselines: (1) **Full Attention** (all 36 layers use FA), (2) **1/2-FA Periodic** (18 layers, every other layer), (3) **1/4-FA Periodic** (9 layers, every 4th layer), (4) **LightTransfer 1/4-FA** (Zhang et al., 2024) (9 layers selected by attention-mass heuristics), and (5) **Naive SWA** (all layers use SWA).

4.2 Main Results

Table 1 presents the main comparison. NLL-Guided 1/4-FA achieves 64.6% accuracy, within 0.4 percentage points of the 1/2-FA Periodic baseline (65.0%) while using only half the full-attention budget (9 vs 18 layers). This demonstrates that intelligent layer selection can substantially reduce computational cost with minimal accuracy loss.

Compared to other 1/4-FA methods, NLL-Guided outperforms the best periodic pattern by 10.4 percentage points (64.6% vs 54.2%), demonstrating that data-driven selection substantially outperforms fixed patterns under the same FA budget. The improvement over LightTransfer is even more pronounced at 26.4 percentage points (64.6% vs 38.2%), indicating that NLL-based scoring provides a fundamentally better signal than attention-pattern heuristics for this task.

Table 1: Accuracy comparison on LongMemEval_24k (500 samples) under different attention configurations. All methods use Qwen3-4B-Thinking-2507 with SWA window=2048, keep_first=10. Best in **bold**, second-best underlined.

Method	FA Layers	Accuracy (%)	Δ vs NLL-Guided
Full Attention	36/36 (all)	**73.0**	+8.4
1/2-FA Periodic	18/36 (every 2nd)	65.0	+0.4
NLL-Guided 1/4-FA (Ours)	9/36 [1,3,9,10,12,13,15,21,34]	64.6	—
1/4-FA Periodic	9/36 [0,4,8,12,16,20,24,28,32]	54.2	−10.4
LightTransfer 1/4-FA	9/36 [2,3,4,5,6,26,29,31,33]	38.2	−26.4
Naive SWA	0/36 (none)	3.2	−61.4

Table 2: Per-task-type accuracy breakdown on LongMemEval_24k. NLL-Guided consistently outperforms LightTransfer across all 6 task types.

Task Type	NLL-Guided (%)	LightTransfer (%)	Δ
knowledge-update	**64.1**	46.2	+17.9
multi-session	**39.1**	20.3	+18.8
single-session-assistant	**96.4**	64.3	+32.1
single-session-preference	**60.0**	46.7	+13.3
single-session-user	**87.1**	50.0	+37.1
temporal-reasoning	**66.2**	32.3	+33.9

4.3 Per-Task Analysis

Table 2 shows the per-task breakdown comparing NLL-Guided and LightTransfer. NLL-Guided outperforms LightTransfer on all six task types, with improvements ranging from 13.3 to 37.1 percentage points. The largest gains appear on single-session-user (+37.1pp) and temporal-reasoning (+33.9pp), suggesting that NLL-guided selection particularly benefits tasks requiring precise long-range information retrieval.

4.4 De-confounding Analysis

A potential concern is that the NLL signal might reflect generic layer sensitivity rather than long-range attention needs. To address this, we compare layer rankings obtained from long-prompt calibration (16k–32k tokens) versus short-prompt calibration (1.5k tokens, within the SWA window where SWA and FA are equivalent).

Figure 2 shows the comparison. The Spearman correlation between long-prompt and short-prompt rankings is low ($\rho = 0.306$, $p = 0.069$), and only 3 of 9 selected layers overlap (Jaccard similarity = 0.2). Furthermore, long-prompt Δ-NLL values are 85.6× larger in magnitude than short-prompt values. These results confirm that the NLL signal is specific to long-range attention needs, not generic layer sensitivity.

4.5 Layer Selection Patterns

Figure 3 visualizes the per-layer Δ-NLL scores. The selected layers [1, 3, 9, 10, 12, 13, 15, 21, 34] naturally span early (1, 3), middle (9–15, 21), and late (34) depths without requiring explicit stratification. Layer 15 shows the highest Δ-NLL (0.011), followed by layers 9, 13, and 21. This non-periodic, data-driven pattern differs fundamentally from fixed periodic selections and suggests that different layers serve distinct roles in long-range information flow.

4.6 Calibration Stability

We analyze the stability of layer selection with respect to calibration set size. Using 16 examples instead of 64 yields a Jaccard similarity of 0.64 with the 64-example selection, with 7 of 9 layers overlapping. Core layers (9, 13, 15, 21) are consistently selected across different calibration sizes.

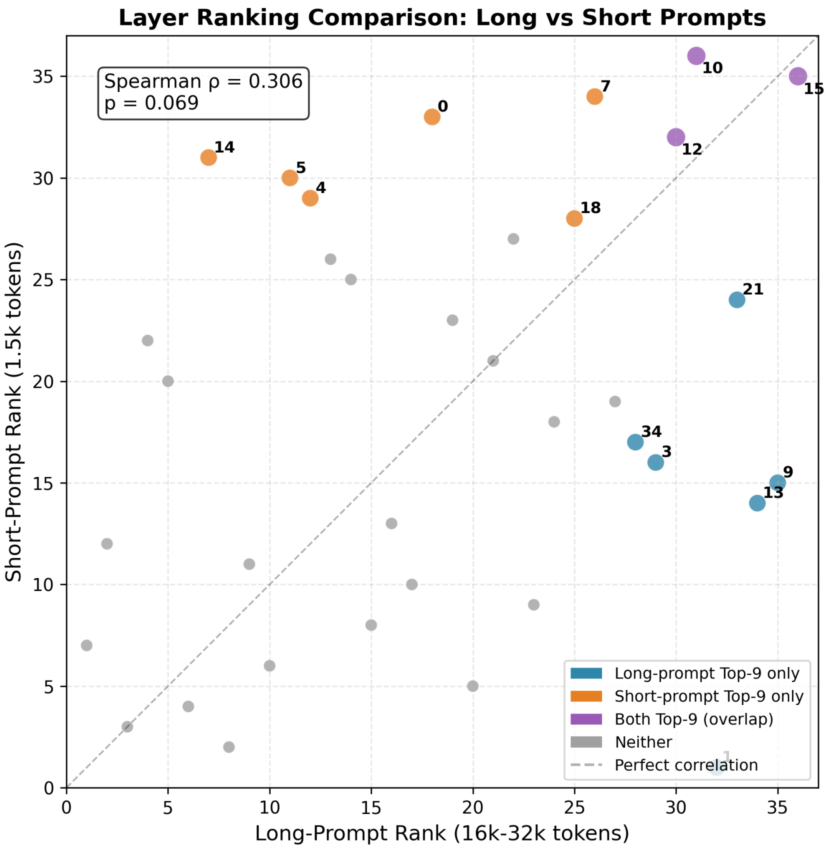

Figure 2: Layer ranking comparison between long-prompt (16k–32k tokens) and short-prompt (1.5k tokens) calibration. Low correlation (Spearman $\rho = 0.306$) and minimal overlap (Jaccard=0.2) confirm the NLL signal is specific to long-range attention needs.

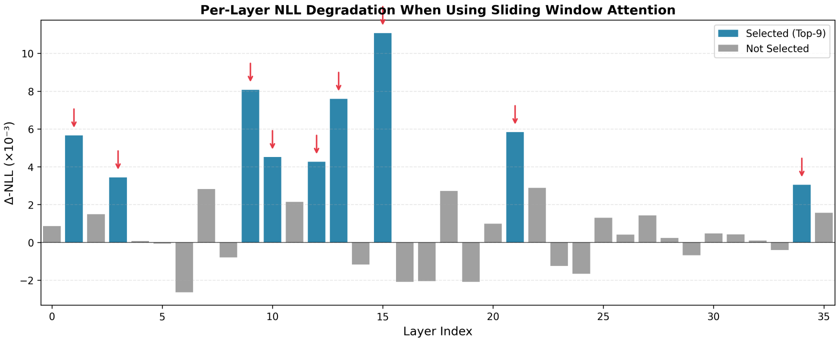

Figure 3: Per-layer NLL degradation (Δ-NLL) when using SWA instead of FA. Blue bars indicate the 9 layers selected for full attention. The selected layers span early, middle, and late depths with a non-periodic pattern.

Downstream accuracy with 16-example calibration is 62.2%, a modest 2.4 percentage point drop from the 64-example result (64.6%), but still substantially above the periodic baseline (54.2%). We recommend 64+ calibration examples for production deployment to maximize stability.

5 Conclusion

We presented NLL-guided layer selection, a principled, training-free method for identifying which layers should retain full attention in hybrid sliding-window attention models. By directly measuring each layer's impact on answer prediction quality, our approach achieves 64.6% accuracy with only 1/4 full-attention layers, matching the 1/2-FA periodic baseline (65.0%) while halving the computational budget. The method outperforms fixed periodic patterns by 10.4 percentage points and attention-based heuristics by 26.4 percentage points.

Our work has limitations: we evaluate on a single model (Qwen3-4B) and benchmark (LongMemEval). Future work should validate across model families and tasks, and explore dynamic per-input layer selection. Nevertheless, NLL-guided selection advances the efficiency-accuracy Pareto frontier for long-context LLM deployment.

References

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

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *ArXiv*, abs/2307.08691, 2023.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Haotong Zhang, and Ion Stoica. *Efficient Memory Management for Large Language Model Serving with PagedAttention*. 2023.

Yanhong Li, Songlin Yang, Shawn Tan, Mayank Mishra, Rameswar Panda, Jiawei Zhou, and Yoon Kim. Distilling to hybrid attention models via kl-guided layer selection. *ArXiv*, abs/2512.20569, 2025.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr F. Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *ArXiv*, abs/2404.14469, 2024.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *ArXiv*, abs/2406.10774, 2024.

Gemma Team, Morgane Riviere, et al. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and I. Polosukhin. Attention is all you need. pp. 5998–6008, 2017.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and A. Eshaghi. Beyond the limits: A survey of techniques to extend the context length in large language models. pp. 8299–8307, 2024.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *ArXiv*, abs/2410.10813, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ArXiv*, abs/2309.17453, 2023.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025.

Yijiong Yu, Jiale Liu, Qingyun Wu, Huazheng Wang, and Ji Pei. Swaa: Sliding window attention adaptation for efficient long-context llms without pretraining. 2025.

M. Zaheer, Guru Guruganesh, Kumar Avinava Dubey, J. Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. *ArXiv*, abs/2007.14062, 2020.

Xuan Zhang, Cunxiao Du, Chao Du, Tianyu Pang, Wei Gao, and Min Lin. Lighttransfer: Your long-context llm is secretly a hybrid model with effortless adaptation. *Trans. Mach. Learn. Res.*, 2025, 2024.

Zhenyu (Allen) Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *ArXiv*, abs/2306.14048, 2023.

A Implementation Details

A.1 Calibration Data

We use 64 long-context examples sampled from LongAlign-10k and fusang-v1-filtered datasets, with prompt lengths between 16k and 32k tokens. Examples are selected to have answer lengths of at least 20 tokens to ensure meaningful NLL computation.

A.2 Scoring Procedure

For each of the 36 layers, we compute the mean NLL on answer tokens under two conditions: (1) all layers use SWA during prefill, and (2) the target layer uses FA while others use SWA. The difference gives the per-layer Δ-NLL score. We use teacher forcing with no gradient computation, enabling efficient calibration.

A.3 SWAA Configuration

Following the SWAA protocol, we use: sliding window size = 2048, keep-first = 10 (attention sinks), and full-attention decode enabled. Generation uses vLLM with enforce_eager=True, temperature=0, and max_completion_len=10000.