BMAttn: Block-Aligned Mixed-Precision Attention Quantization for LLM Inference

Zining Wang[*,1,2]
wangzining@buaa.edu.cn

Haojie Duanmu[*,4,5]
duanmuhaojie@sjtu.edu.cn

Zhihang Yuan[6]
yuanzhihang@pku.edu.cn

Fanliu Kong[8]
220180780@seu.edu.cn

Ruihao Gong[1,2,7]
gongruihao@buaa.edu.cn

Jinyang Guo[1,3]
jinyangguo@buaa.edu.cn

Xianglong Liu[†,1,2]
xlliu@buaa.edu.cn

Abstract

Long-context deployment of large language models is bottlenecked by the quadratic cost of self-attention. Existing acceleration methods struggle to balance adaptivity and efficiency: uniform sparsity or quantization ignores the highly non-uniform importance of attention scores, while fine-grained adaptive schemes often break GPU efficiency through irregular memory access. We present BMAttn, a block-aligned mixed-precision attention framework that resolves this trade-off by combining importance-aware compression with hardware-friendly execution. BMAttn partitions attention maps into high-precision, low-precision, and skipped regions, and parameterizes their boundaries as affine functions of sequence length so that one calibration can generalize across contexts. To calibrate these boundaries, we introduce an IPW-based saliency objective that removes the empirical short-range bias of raw attention mass. We show that IPW is the only distance-based reweighting (up to a constant factor) that yields a distance-uniform retention constraint, and we derive a corresponding per-row attention-output approximation bound. We further pair the calibration with a layer-adaptive regularizer that aligns retention targets with layer sensitivity. Across six benchmarks and three LLM families, BMAttn matches FlashAttention-2 within 0.5 percentage points while delivering up to 3.26× attention-kernel speedup on Qwen2.5-7B-Instruct, substantially improving the efficiency-accuracy trade-off for long-context inference. Our code is available at https://github.com/MercuryB1/BMAttn.

1 Introduction

Recent advancements in Large Language Models (LLMs) have yielded remarkable capabilities across a wide range of applications [Liu et al., 2024, Achiam et al., 2023, Yang et al., 2023]. A prominent trend in this evolution is the dramatic expansion of context windows, with models like Gemini [Comanici et al., 2025] and Kimi [Team et al., 2025] now capable of processing sequences exceeding one million tokens. However, this expansion is fundamentally constrained by the underlying self-attention mechanism [Vaswani et al., 2017]. Its computational complexity and

[1]State Key Laboratory of Complex & Critical Software Environment, [2]School of Computer Science and Engineering, Beihang University, [3]School of Artificial Intelligence, Beihang University, [4]Shanghai Jiao Tong University, [5]Shanghai AI Laboratory, [6]Peking University, [7]Beijing Academy of Artificial Intelligence. [8]Southeast University. *Equal contribution. †Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

memory footprint scale quadratically ($O(L^2)$) with the input sequence length L, posing a prohibitive bottleneck to the scalability and practical deployment of these powerful models.

Existing approaches to mitigate this cost, such as sparse attention [Zhang et al., 2025b, Fu et al., 2024, Gao et al., 2024, Xu et al., 2025, Xiao et al., 2024] and quantization [Zhang et al., 2024b,a], typically rely on uniform compression strategies. For instance, standard sparse attention often enforces a rigid keep-or-discard policy, while conventional quantization applies a homogeneous bit-width across all attention connections. These approaches are intrinsically misaligned with the non-uniform nature of attention. Empirical observations consistently show that attention patterns are highly heterogeneous: some heads specialize in local, high-precision interactions, while others aggregate global context that is robust to coarse approximation. A uniform strategy fails to adapt to this variance, allocating too many bits to negligible connections or pruning critical ones, leading to a suboptimal trade-off between efficiency and accuracy

This reveals a critical need for a framework capable of importance-aware precision allocation. However, the central challenge lies in implementing such fine-grained adaptivity without incurring the system-level penalties of irregular memory access. Naive token-level mixed precision introduces warp divergence and scattered memory operations, which can easily nullify the theoretical FLOPs reduction on modern GPUs. Therefore, a practical solution must bridge the gap between algorithmic adaptivity and hardware regularity.

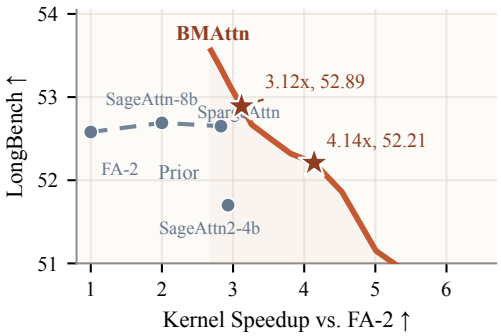


Figure 1: Efficiency-accuracy trade-off on Qwen2.5-7B, building a new Pareto frontier.

To resolve this tension, we introduce **Block-Aligned Mixed-Precision Attention (BMAttn)**, a novel framework that co-designs an adaptive precision scheme with a hardware-friendly computational structure. The core principle of BMAttn is to partition the attention map into three distinct zones based on causal distance: a high-precision (HP) zone for critical short-range interactions, a low-precision (LP) zone for retrievable long-range context, and a sparse (SP) zone where negligible connections are pruned. Crucially, to ensure hardware efficiency, this partitioning is block-aligned. By enforcing precision boundaries at the block level, BMAttn creates a regular, staircase-shaped computational pattern that is highly amenable to GPU acceleration and compatible with optimized kernels like FlashAttention [Dao, 2023].

Beyond structural efficiency, BMAttn incorporates dynamic adaptability to handle diverse inputs. Instead of using static, hand-tuned thresholds, we propose an **affine window mechanism** that parameterizes zone boundaries as affine functions of the sequence length. This allows each attention head to flexibly adjust its attention span during inference. Furthermore, to determine the optimal parameters, we introduce a **saliency-weighted calibration** method that corrects for the locality bias of standard metrics, alongside a **layer-adaptive retention regularizer** that enforces stricter precision targets on sensitive shallow layers.

In summary, our contributions are as follows:

- We introduce BMAttn, a framework that unifies mixed-bit quantization with a block-aligned structure and affine windowing to achieve both fine-grained adaptivity and hardware-friendly execution.
- We propose a saliency-weighted calibration based on Inverse-Propensity Weighting (IPW) and provide two theoretical results. Theorem G.1 shows that IPW is the unique (up to scale) distance reweighting whose retention constraint is distance-uniform. Proposition G.7 lifts this constraint-level characterization to a per-row attention-output approximation bound, with the bound spread evenly across distance buckets and independent of the empirical attention density. We pair the calibration with a depth-clipped layer-adaptive regularizer with explicit feasibility conditions.
- We conduct extensive experiments across three LLMs and six benchmarks (WikiText-2, MMLU, LongBench, InfiniteBench, NIAH, RULER), showing that BMAttn delivers a $3.26\times$

attention-kernel speedup on Qwen2.5-7B while staying within 0.5 pp of FlashAttention-2 on the per-method average, improving the efficiency–accuracy trade-off over state-of-the-art mixed-precision and sparse baselines.

2 Related Works

Algorithmic Optimizations. Approaches to mitigate the quadratic complexity of attention have evolved significantly. Early methods introduced structured sparsity through fixed patterns [Liu et al., 2021, Chu et al., 2021], while subsequent works like StreamingLLM [Xiao et al., 2023] and LongLoRA [Chen et al., 2023] adopted adaptive context management to preserve crucial information. More recently, inference-centric designs such as Minference [Jiang et al., 2024] and SpargeAttention Zhang et al. [2025b] achieve multi-fold speedups by dynamically skipping near-zero entries. Furthermore, to address the KV cache memory bottleneck, methods like DuoAttention [Xiao et al., 2024] and ShadowKV [Sun et al., 2024] distinguish retrieval-critical heads, effectively reducing memory usage by 2–6$\times$ while maintaining long-context accuracy.

System-Level Optimizations. System-level optimizations focus on maximizing hardware utilization through efficient kernels and quantization. Libraries like xFormers [Lefaudeux et al., 2022] provided early modular kernels, while FlashAttention [Dao et al., 2022, Dao, 2023, Shah et al., 2024] revolutionized the field with I/O-aware tiling to minimize HBM access. In parallel, quantization has matured from early BERT-specific works [Kim et al., 2021] to general LLM acceleration. Notably, SageAttention [Zhang et al., 2024b, 2025a] demonstrates that fully quantized INT8 attention can serve as a drop-in replacement for FP16, outperforming standard kernels in both speed and accuracy.

3 Preliminaries

Full attention. Self-attention projects inputs into queries, keys, and values $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{L \times d_k}$, computes scores $\mathbf{S} = \mathbf{Q}\mathbf{K}^\top$, compute attention weight $\mathbf{W} = \mathrm{softmax}(\mathbf{S}/\sqrt{d_k})$, and outputs $\mathbf{O} = \mathbf{W}\mathbf{V}$. The score computation scales as $\mathcal{O}(L^2 d_k)$ and dominates for long sequences.

Windowed attention. Attention computation is reduced by restricting each query q_i to a subset of keys $\mathcal{J}_i \subset \{0, \ldots, L-1\}$ via masking: $\mathbf{S}'_{ij} = \mathbf{S}_{ij}$ if $j \in \mathcal{J}_i$ and $\mathbf{S}'_{ij} = -\infty$ otherwise. A common design uses a fixed attention sink of length s and a local window of size k: $\mathcal{J}_i = \{0, \ldots, s-1\} \cup \{\, j \mid \max(s, i-k+1) \le j \le i \,\}$. This reduces computed scores from L^2 to roughly $L(s+k)$, yielding linear complexity in L.

Quantization. Quantization accelerates matrix multiplication (e.g., $\mathbf{C} = \mathbf{A}\mathbf{B}$) by mapping high-precision matrices to low-precision integers via a quantizer ψ and de-quantizer ψ^{-1}. With $(\delta_A, \hat{\mathbf{A}}) = \psi(\mathbf{A})$ and $(\delta_B, \hat{\mathbf{B}}) = \psi(\mathbf{B})$, we compute $\hat{\mathbf{C}} = \hat{\mathbf{A}}\hat{\mathbf{B}}$ using integer arithmetic and recover $\mathbf{C} \approx \psi^{-1}_{\delta_A, \delta_B}(\hat{\mathbf{C}}) = \hat{\mathbf{C}} \cdot \delta_A \delta_B$. Quantization granularity controls which elements share a scale δ; for symmetric INT8, $\delta_{\text{group}} = \max(|\mathbf{X}_{\text{group}}|)/127$. Beyond per-tensor scaling, per-block quantization partitions matrices into blocks and assigns a scale per block, improving fidelity to local statistics with modest overhead.

For Attention, quantization acts on the matrix-multiplication operands: (i) $\mathbf{Q}\mathbf{K}^\top$ path via quantized $\mathbf{Q}$ and $\mathbf{K}$; (ii) $\mathbf{W}\mathbf{V}$ path via quantized $\mathbf{V}$ and quantized attention weights $\mathbf{W}$. Attention quantization enables faster computation and memory access.

4 Method

We propose **BMAttn**, a Block-aligned Mixed-precision Attention framework that reconciles fine-grained adaptive attention with hardware efficiency (Fig. 2). BMAttn allocates precision heterogeneously across the attention map while enforcing a block-aligned compute pattern that is compatible with optimized kernels such as FlashAttention [Dao, 2023]. High precision (e.g., INT8) is assigned to critical short-range dependencies, low precision (e.g., INT4) to retrievable long-range context, and zero precision (skip) to negligible regions. Precision boundaries are dynamically determined by an affine windowing mechanism so that one offline calibration generalizes across sequence lengths. The

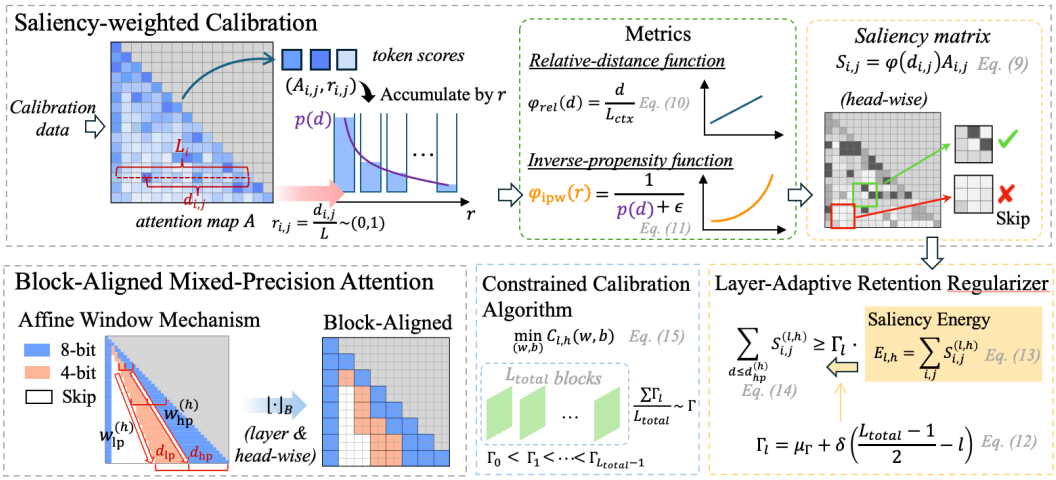


Figure 2: Overview of **BMAttn**. The method consists of two stages: (1) Offline Calibration (Top & Right): We compute a saliency matrix using Inverse-Propensity Weighting (IPW) to correct for locality bias, while a Layer-Adaptive Regularizer sets depth-aware retention targets (Γ_l). These components guide the constrained optimization of affine parameters. (2) Block-Aligned Inference (Bottom-Left): Based on the calibrated parameters, the Affine Window Mechanism dynamically partitions the attention map into High-Precision (8-bit), Low-Precision (4-bit), and Sparse zones. These boundaries are discretized into blocks to ensure hardware-friendly execution.

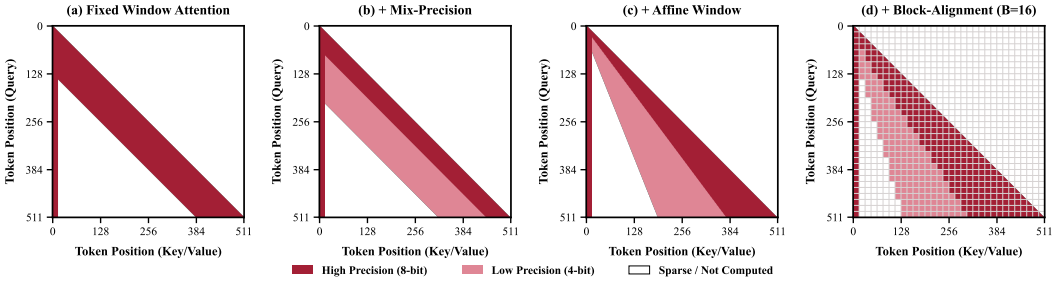


Figure 3: Comparison of different attention window schemes. (a) Window Attention: Fixed-size sliding window. (b) Constant Precision Tiers: High and low precision regions. (c) Affine Window: Adaptive precision boundaries based on context length. (d) Block-Alignment: Attention is divided into contiguous blocks, optimizing hardware efficiency.

affine parameters are calibrated using a saliency-weighted metric that corrects for locality bias, paired with a layer-adaptive regularizer that respects the model's feature hierarchy.

4.1 Block-Aligned Mixed-Precision Attention

Motivation. Scaling the context window of LLMs is fundamentally constrained by the $O(L^2)$ cost of self-attention in computation and memory movement. Two empirical observations make uniform precision or sparsity assignments intrinsically suboptimal. First, attention mass and its semantic criticality decay with causal distance: short-range links encode fine-grained lexical and syntactic cues, while mid- or long-range links contribute broader discourse-level signals. Second, different heads specialize in distinct ranges, with some focused on local interactions and others on global context. A single bit-width or sparsity pattern across the entire map therefore wastes compute on unimportant regions or harms accuracy by underrepresenting crucial ones.

Fine-grained adaptivity alone is not enough. Token-level mixed precision and unaligned per-row cutoffs introduce non-coalesced memory accesses and warp divergence that nullify theoretical savings on modern GPUs. Fixed, hand-tuned distance cutoffs further fail to generalize across context lengths, since a window calibrated at one length can be over- or under-aggressive at another.

Table 1: Evidence of the **Sensitivity Gap** on Qwen2.5-7B. Each row corresponds to a global retention ratio (the fraction of attention mass kept by the windowed approximation). All four standard metrics remain ostensibly stable as the retention ratio decreases, while perplexity degrades catastrophically; only IPW (our reweighted coverage $\rho_{\varphi^\star}$) tracks PPL.

Ratio	Cos Sim	RMSE	Relative L1	ASC	**IPW**	Ppl.
95%	99.92%	0.1186	0.0887	0.129	0.9123	**44.3892**
90%	99.95%	0.0881	0.0616	0.103	0.8880	**21.4239**
80%	99.98%	0.0594	0.0368	0.087	0.7955	**16.9872**
70%	99.99%	0.0303	0.0181	0.056	0.6131	**14.2538**

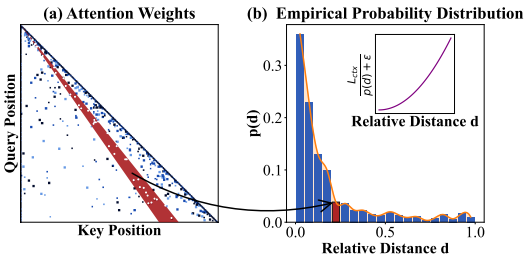


Figure 4: Derivation of **IPW**. We estimate empirical density $p(d)$ via bucketing and compute weighting function $\varphi(d) \propto 1/p(d)$ to compensate for sparsity at long distance.

Three-zone decomposition. As shown in Fig. 3, we partition each head's causal attention map into three monotone bands along distance: a high-precision (HP) zone for short-range, high-saliency links (INT8); a low-precision (LP) zone for mid- or long-range links that tolerate lower bit-width (INT4); and a sparse (SP) zone where negligible long-range links are skipped (zero bits). Sink tokens [Xiao et al., 2023] are unconditionally assigned to HP regardless of distance, stabilizing retrieval over extremely long contexts.

Affine zone boundaries. Let L_{seq} be the total sequence length, L_{sink} the number of protected sink tokens, and $L_{\text{ctx}} = L_{\text{seq}} - L_{\text{sink}}$ the effective context length. For each head h, we parameterize the HP and LP boundaries as affine functions of L_{ctx}:

$$d_{\text{hp}}^{(h)} = w_{\text{hp}}^{(h)} \cdot L_{\text{ctx}} + b_{\text{hp}}^{(h)}, \qquad d_{\text{lp}}^{(h)} = w_{\text{lp}}^{(h)} \cdot L_{\text{ctx}} + b_{\text{lp}}^{(h)}. \tag{1}$$

The slope $w_{\cdot}^{(h)}$ captures the head's sensitivity to context scaling, while the bias $b_{\cdot}^{(h)}$ represents its local receptive field. Both are calibrated offline and remain fixed during inference. Subject to $0 \le d_{\text{hp}}^{(h)} \le d_{\text{lp}}^{(h)} \le L_{\text{ctx}}$, the thresholds partition the causal attention map (with causal distance $d_{i,j} = i - j$) into the three zones introduced above.

Block-aligned execution. To bridge continuous affine boundaries to GPU tile structure, we snap the thresholds to the nearest lower block boundary of size B (e.g., 64 or 128):

$$D_{\text{hp}}^{(h)} = \lfloor d_{\text{hp}}^{(h)} \rfloor_B, \qquad D_{\text{lp}}^{(h)} = \lfloor d_{\text{lp}}^{(h)} \rfloor_B, \tag{2}$$

producing a regular staircase of homogeneous tiles. The resulting zones are

$$\mathcal{Z}_{\text{hp}}^{(h)} = \{(i,j) \mid j < i,\ (j < L_{\text{sink}}) \ \vee \ (0 \le d_{i,j} \le D_{\text{hp}}^{(h)})\}, \tag{3}$$

$$\mathcal{Z}_{\text{lp}}^{(h)} = \{(i,j) \mid j < i,\ D_{\text{hp}}^{(h)} < d_{i,j} \le D_{\text{lp}}^{(h)}\}, \tag{4}$$

$$\mathcal{Z}_{\text{sp}}^{(h)} = \{(i,j) \mid j < i,\ d_{i,j} > D_{\text{lp}}^{(h)}\}. \tag{5}$$

Each tile is assigned a uniform precision tier (HP: 8-bit, LP: 4-bit, SP: zero-bit / skipped) that the QK and PV matmuls share, eliminating warp divergence within the kernel. The exact per-tensor quantization scheme (per-block versus per-channel, and the floating-point format used for the value path) is part of the implementation rather than the method, and is detailed in Section 5.1 and Appendix F. Because the zone boundaries are fully determined by the two block indices $(D_{\text{hp}}^{(h)}, D_{\text{lp}}^{(h)})$, we map them directly to FlashAttention loop bounds and avoid materializing an $O(L^2)$ boolean mask.

4.2 Saliency-Weighted Calibration

4.2.1 Saliency-Weighted Metric

Motivation. Standard reconstruction metrics (MSE, cosine similarity, ASC) suffer from severe locality bias: attention's heavy tail lets short-range tokens dominate the loss, so these metrics remain near-saturated even as model perplexity collapses (Table 1). We refer to this disconnect as the *sensitivity gap*: standard metrics fail to detect the loss of rare but semantically critical long-range dependencies, motivating a weighted calibration objective.

Distance-uniform retention via IPW. The right calibration target for long-context retrieval is *distance-uniform retention*: every distance bucket should be equally protected by the saliency-retention constraint, regardless of the empirical concentration $p(d)$. We realize this with Inverse-Propensity Weighting (IPW), reweighting raw attention $\mathbf{W}$ by the inverse density:

$$S_{i,j}^{(l,h)} = \underbrace{\frac{L_{\text{ctx}}}{p^{(l,h)}(d_{i,j}) + \epsilon}}_{\varphi_{\text{IPW}}(d)} \cdot \mathbf{W}_{i,j}^{(l,h)}, \tag{6}$$

where $p^{(l,h)}(d)$ is estimated via distance bucketing (Fig. 4) and ϵ is a numerical-stability constant.

Theoretical justification. Under a fixed distance bucketization, IPW is the unique distance reweighting (up to scale) whose retention constraint is distance-uniform (Theorem G.1, Appendix G); without IPW, the heavy-tailed p of pretrained LLMs collapses the constraint onto a short prefix and recovers the locality bias of Table 1. We further lift this characterization to a calibration-time per-row attention-output error bound for the SP truncation under bounded value norms (Proposition G.7 and Corollaries G.8–G.9 in Appendix G.6). The bound covers the SP truncation only; it does not account for HP or LP quantization noise, and it is a constraint-level guarantee rather than a task-loss bound. End-to-end accuracy retention is established empirically in Section 5.2.

4.2.2 Layer-Adaptive Retention Regularizer

Motivation. While IPW addresses locality bias within attention heads, it treats all network layers uniformly. This overlooks the hierarchical nature of LLM feature processing: shallow layers typically encode fragile, high-fidelity lexical information, whereas deeper layers process abstract semantic representations that are often redundant and robust to approximation [Huang et al., 2025]. A monolithic retention target is suboptimal, since it risks over-compressing critical early layers while failing to exploit the higher compressibility of deeper ones.

Layer-adaptive retention budget. We formulate the layer-specific budget Γ_l as a clipped linear function of depth, parameterized by a global retention mean $\mu_\Gamma \in (0, 1]$ and a decay factor $\delta \geq 0$:

$$\Gamma_l = \text{clip}_{(0,1]}\left[\mu_\Gamma + \delta \cdot \left(\frac{L_{\text{total}} - 1}{2} - l\right)\right], \tag{7}$$

where L_{total} is the total number of layers indexed by $l \in \{0, \ldots, L_{\text{total}} - 1\}$. A positive δ transfers the budget from robust deep layers to sensitive shallow ones, realizing a "protect-the-foundation" policy. The schedule is feasible (no clipping triggered) iff

$$\delta \;\leq\; \delta_{\max}(\mu_\Gamma, L_{\text{total}}) \;:=\; \frac{2\min(\mu_\Gamma, 1 - \mu_\Gamma)}{L_{\text{total}} - 1}, \tag{8}$$

within which the layer-averaged retention equals μ_Γ exactly; outside, clipping perturbs the average and we report the effective retention separately. All experiments operate in the feasible region (e.g., the default $(\mu_\Gamma, \delta) = (0.8, 0.01)$ is well below $\delta_{\max} \approx 0.013$ for 32-layer Qwen2.5-7B). Aggressive-sweep settings with μ_Γ close to 1 may marginally clip at the top boundary, with the effective retention reported in Appendix D (deviations below 0.5 pp).

4.2.3 Constrained Calibration Algorithm

Given the IPW saliency and the per-layer budget Γ_l, we calibrate the affine parameters $\Theta = \{(w_z^{(l,h)}, b_z^{(l,h)}) : z \in \{\text{hp}, \text{lp}\}\}$ by minimizing the expected window size subject to the retention constraints. Let $E_{l,h} = \sum_{\mathbf{X} \in \mathcal{D}_{\text{calib}}} \sum_{i,j} S_{i,j}^{(l,h)}(\mathbf{X})$ be the total saliency energy of head (l, h), and let $\gamma \in (0, 1)$ specify the portion of retained energy reserved for the HP zone. For each (l, h),

$$\min_{w_\cdot^{(l,h)},\, b_\cdot^{(l,h)}} \quad \mathbb{E}_{L \sim \mathcal{D}_{\text{calib}}}\left[d_{\text{hp}}^{(l,h)}(L) + d_{\text{lp}}^{(l,h)}(L)\right] \tag{9}$$

$$\text{s.t.} \quad \sum_{(i,j) \in \mathcal{Z}_{\text{hp}}^{(l,h)} \cup \mathcal{Z}_{\text{lp}}^{(l,h)}} S_{i,j}^{(l,h)} \;\geq\; \Gamma_l \cdot E_{l,h}, \qquad \text{(Total Retention)}$$

$$\sum_{(i,j) \in \mathcal{Z}_{\text{hp}}^{(l,h)}} S_{i,j}^{(l,h)} \;\geq\; \gamma \cdot \Gamma_l \cdot E_{l,h}. \qquad \text{(HP Constraint)}$$

The first constraint requires HP and LP together to retain at least a Γ_l fraction of the saliency energy; the second reserves a γ fraction of that budget for HP. Among all feasible boundaries, we choose the one with the smallest expected window size, which serves as a proxy for retained blocks and kernel work. For each (l, h), we grid-search the four affine scalars and select the smallest feasible configuration; this is tractable because retained saliency increases monotonically with window size. The full algorithm is given in Appendix A.

5 Experiments

5.1 Setup

Models. We evaluate `BMAttn` across three representative large language models (LLMs): Qwen2.5 (7B) [Qwen et al., 2025], Llama3 (8B) [Dubey et al., 2024], and GLM4 (9B) [GLM et al., 2024]. These models span different architectures and training regimes, providing a diverse testbed for assessing the performance and efficiency of our method.

Datasets and Metrics. Our evaluation spans both standard and long-context benchmarks: WikiText-2 [Merity et al., 2016] for language modeling perplexity, MMLU [Hendrycks et al., 2020] for multi-task understanding, LongBench [Bai et al., 2023] and InfiniteBench [Zhang et al., 2024c] for long-context reasoning, RULER [Hsieh et al., 2024] for reasoning and reading comprehension, and NIAH [Kamradt, 2023] for retrieval ability. Metrics include perplexity (lower is better), accuracy, and task-specific scores, depending on the nature of the task.

Baselines. We compare `BMAttn` with several state-of-the-art attention mechanisms, including FlashAttention-2 [Dao, 2023], SageAttention [Zhang et al., 2024b], SageAttention2 [Zhang et al., 2024a], and SpargeAttention [Zhang et al., 2025b]. These baselines are widely regarded as strong implementations in the efficient attention space and provide a robust comparison across both accuracy and efficiency.

Implementation Details. For quantization, we employ a hybrid scheme where Q, K, and W are quantized per block, while V is quantized per channel. This approach ensures efficient block-aligned execution while preserving accuracy in the output representations. In all experiments, we set the average retention rate $\mu_\Gamma = 0.7$, decay factor $\delta = 0.01$, and $\gamma = 0.9$ (these satisfy the feasibility condition $\delta \leq \delta_{\max}$ in Eq. (8)). For the saliency metric we report results using IPW. We evaluate on NVIDIA A800-80G (accuracy benchmarks, attention-kernel speedup) and RTX 4090 (end-to-end latency). The code is built upon the open-source SageAttention2 implementation. Kernel implementation is detailed in Appendix F.

Calibration Protocol. The IPW propensity $p^{(l,h)}(d)$ and total saliency energy $E_{l,h}$ are estimated on WikiText [Merity et al., 2016]. Causal distances (excluding L_{sink}=128 sink tokens) are partitioned into contiguous distance buckets $\mathcal{B}$=64 as described in Appendix H. For each (layer, head) the constrained optimization (Eq. (9)) is solved by grid search over the four affine scalars. Per-model calibration peaks at 14.4 GB GPU memory and completes in 1m15s on Qwen2.5-7B-Instruct. The sequence length is 2048 and the samples amount is 512.

5.2 Accuracy

In Table 2, we present the results comparing `BMAttn` with the baselines. The findings highlight that `BMAttn` achieves competitive accuracy with state-of-the-art methods, while delivering substantial improvements in computational efficiency across all the models and datasets. Specifically, for Qwen2.5-7B-Instruct, `BMAttn` maintains near-lossless accuracy compared to the FlashAttention-2 and SageAttention-8b, while achieving a 3.26× speedup. In addition, `BMAttn` outperforms SageAttention2-4b, both in terms of accuracy (52.67 vs. 51.70 on LongBench) and speedup (3.26× vs. 2.93×). These results

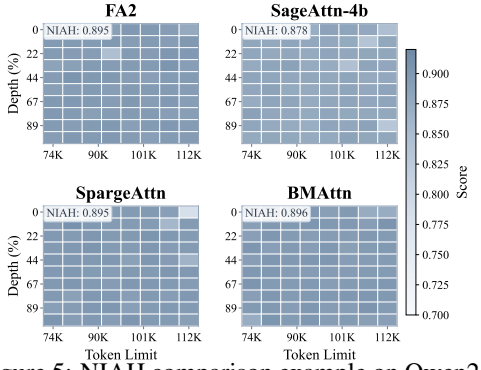


Figure 5: NIAH comparison example on Qwen2.5.

Table 2: Accuracy comparison of BMAttn with baseline methods across 3 LLMs and 6 benchmarks. We compute the score proportion relative to the baseline (*i.e.*, FlashAttention-2) across benchmarks, and then compute the average value in the "(Avg. (%))" column.The best and second-best results are highlighted in **bold** and underlined formats, respectively. More accuracy reults can be found in Appendix E.

Method	WikiText (Ppl.)↓	MMLU↑	LongBench↑	InfiniteBench↑	NIAH↑	RULER↑	Avg. (%)	Speedup
Qwen2.5-7B-Instruct [Qwen et al., 2025]								
FlashAttention-2 [Dao, 2023]	7.458	0.717	52.58	0.285	0.882	94.05	100.0	1.00×
SageAttention-8b [Zhang et al., 2024b]	7.463	**0.716**	**52.69**	0.277	0.878	93.59	99.26	2.00×
SageAttention2-4b [Zhang et al., 2024a]	7.582	0.702	51.70	0.261	0.870	88.67	96.15	2.93×
SpargeAttention [Zhang et al., 2025b]	7.465	0.711	52.65	0.286	**0.885**	93.98	99.98	3.20×
BMAttn (5.75 bits)	**7.461**	**0.716**	52.67	**0.288**	**0.885**	**94.01**	**100.28**	**3.26**×
Llama3.1-8B-Instruct [Dubey et al., 2024]								
FlashAttention-2 [Dao, 2023]	7.217	0.629	54.09	0.389	0.899	91.27	100.0	1.00×
SageAttention-8b [Zhang et al., 2024b]	7.223	**0.627**	**54.07**	0.380	0.900	90.00	99.21	2.15×
SageAttention2-4b [Zhang et al., 2024a]	7.461	0.598	52.90	0.369	0.874	85.24	95.67	2.96×
SpargeAttention [Zhang et al., 2025b]	7.221	0.626	54.05	**0.390**	0.895	91.10	99.81	**3.12**×
BMAttn (5.75 bits)	**7.220**	**0.627**	54.00	0.388	**0.901**	**91.15**	**99.87**	3.11×
GLM-4-9B-Chat [GLM et al., 2024]								
FlashAttention-2 [Dao, 2023]	11.937	0.682	53.53	0.312	0.903	92.33	100.0	1.00×
SageAttention-8b [Zhang et al., 2024b]	11.997	0.680	53.19	0.309	0.900	**92.28**	99.54	1.96×
SageAttention2-4b [Zhang et al., 2024a]	12.219	0.643	50.38	0.298	0.889	88.96	95.745	2.95×
SpargeAttention [Zhang et al., 2025b]	**11.959**	0.675	53.20	0.311	**0.905**	92.22	99.63	3.18×
BMAttn (5.75 bits)	11.965	**0.681**	**53.26**	**0.314**	0.902	92.26	**99.96**	**3.20**×

validate the effectiveness of `BMAttn` in balancing high accuracy with significant efficiency gains, demonstrating its potential as a robust solution for large language models. Fig. 5 also illustrates that `BMAttn` outperform in retrieving tasks.

5.3 Efficiency

We measure efficiency along two part: *end-to-end* model latency and *attention-kernel* throughput. Both are evaluated under the same lossless configuration as Section 5.2. More Efficiency results are detailed in Appendix E.

Table 3: End-to-end model latency ↓ (ms) and speedup ↑ across sequence lengths on RTX 4090. Best results in **bold**.

Method	SeqLen											
	1k		2k		4k		8k		16k		32k	
	Latency	Speedup	Latency	Speedup	Latency	Speedup	Latency	Speedup	Latency	Speedup	Latency	Speedup
FA2	3.091	-	6.390	-	13.341	-	27.881	-	63.176	-	151.772	-
SA-8b	3.058	1.01×	6.265	1.02×	12.872	1.03×	26.148	1.06×	56.143	1.12×	124.496	1.21×
SA2-4b	3.036	1.01×	6.221	1.02×	12.738	1.04×	25.538	1.09×	53.718	1.17×	115.556	1.31×
Sparge	4.729	0.65×	7.604	0.84×	14.433	0.92×	26.943	1.03×	55.627	1.13×	116.481	1.30×
BMAttn	**3.033**	**1.01**×	**6.210**	**1.02**×	**12.699**	**1.05**×	**25.413**	**1.09**×	**53.268**	**1.18**×	**113.949**	**1.33**×

Lossless End-to-End Latency. Table 3 reports end-to-end wall-clock latency on RTX 4090, where BMAttn consistently achieves the lowest latency across all sequence lengths (1k–32k). At long contexts (32k), BMAttn delivers a peak **1.33× speedup**, surpassing both the quantization-based SageAttention (1.31×) and the sparse SpargeAttention (1.30×). Crucially, BMAttn eliminates the "optimization tax" often seen in dynamic sparse methods: while SpargeAttention suffers significant regression at short lengths (0.65× at 1k) due to indexing overhead, BMAttn maintains positive acceleration ($> 1.0\times$) throughout, proving its viability as a robust, general-purpose drop-in replacement.

Attention Kernel Speedup. To isolate the attention component from the rest of the model, we additionally measure the practical attention-only kernel throughput on RTX4090 (Fig. 6) using `headdim=128` and causal masking [Vaswani et al., 2017].. BMAttn delivers a 3.26× kernel speedup over FlashAttention-2 at 32k, corresponding to 1.63× over SageAttention-8b and 1.10× over SageAttention2-4b. The acceleration grows with sequence length, indicating that BMAttn exploits long context effectively through block-aligned computation and SP-zone block skipping.

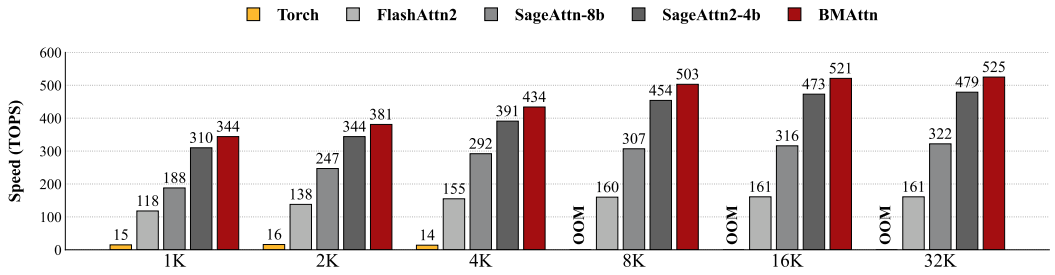


Figure 6: Lossless attention-kernel throughput of **BMAttn** vs. baselines on RTX4090, relative to FlashAttention-2 across sequence lengths.

Table 4: Ablation on window design.

Seqlen	Fixed Window		Affine Window	
	RULER	Speedup	RULER	Speedup
4K	94.05	1.91×	94.01	3.26×
8K	92.39	3.12×	92.48	3.14×
16K	89.15	3.45×	91.89	3.24×
32K	44.69	5.45×	88.69	3.26×
64K	23.54	9.11×	83.20	3.29×
128K	6.88	19.81×	73.35	3.33×

Table 5: Ablation on SWM and LRR.

Model	SWM	LRR	LongBench ↑
Qwen2.5-7B-Instruct	✗	✗	46.53
	✗	✓	50.06
	✓	✗	52.01
	✓	✓	**52.67**
Llama3.1-8B-Instruct	✗	✗	48.66
	✗	✓	51.38
	✓	✗	53.04
	✓	✓	**54.00**

5.4 Ablation Study

Affine Window Ablation. To demonstrate the importance of affine window mechanism, we compare our affine window against a fixed-window baseline, whose window sizes are calibrated at 8k tokens and then held constant. Table 4 reveals the fundamental flaws of this rigid strategy, which is only effective when the inference context closely matches the calibration context. On short sequences (4k), the fixed window is overly conservative; it preserves a large region at high precision that contains little valuable information, leading to suboptimal compression and thus limited speedup (1.91× vs. 3.26×). Conversely, on long sequences (128k), the window becomes severely restrictive. It fails to cover essential long-range dependencies, causing a catastrophic accuracy collapse (RULER score of 6.88). In contrast, our affine window gracefully adapts to all scenarios, maintaining robust accuracy while delivering a consistent ∼3.3× speedup. These results confirm that dynamic affine scaling is essential for building a generalizable and efficient model.

Components Ablation. The ablation in Table 5 clearly demonstrates the necessity of both the saliency-weighted metric (SWM) and the layer-adaptive retention regularizer (LRR). Without LRR, the model suffers a severe drop in LongBench score (46.53), confirming that uniform retention targets are ill-suited given the heterogeneous sensitivity of different layers. Introducing LRR alone already restores performance to 50.06, as depth-aware allocation prevents shallow layers from collapsing under aggressive pruning. SWM provides complementary benefits: by reweighting attention calibration toward semantically crucial long-range pairs, it recovers additional accuracy (52.01). When combined, SWM+LRR achieves the best results, reaching 52.67 on Qwen2.5-7B and 54.00 on Llama3.1-8B. This synergy reflects their distinct yet aligned roles—SWM corrects locality bias at the head level, while LRR counteracts asymmetry across depth—together producing a balanced strategy that generalizes across architectures.

Calibration Cost. In our method, the calibration process involves offline optimization of affine window parameters for each attention head, which requires a one-time computation. On the Qwen2.5-7B-Instruct model, the peak GPU memory usage during calibration is 14,362 MB, and the total calibration time is 1 minute and 15 seconds. This calibration cost is a one-time expense that occurs during the model setup phase and does not affect the subsequent inference time, which remains efficient due to the optimizations in precision and sparsity handling during the attention computation. This calibration time and memory usage are reasonable given the significant improvements in computational efficiency and accuracy that `BMAttn` delivers during inference.

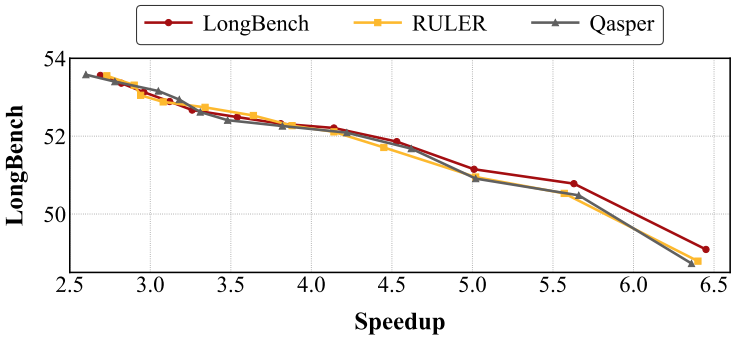


Figure 7: Longbench results of different calibration data.

Calibration Data Ablation. Fig. 7 illustrates the performance of our method using different calibration datasets (LongBench, RULER, and Qasper), with LongBench as the baseline. At low compression rates, the performance remains consistent across all calibration datasets, indicating that our method is robust to the choice of calibration data. However, as compression increases, we observe a slight degradation in performance when using RULER or Qasper compared to LongBench. This degradation is minor, and the overall performance remains competitive. These results show that while LongBench provides a slight advantage at higher compression rates, our method remains robust and performs well across different calibration datasets, with minimal impact on performance at lower compression levels.

6 Conclusion

We presented `BMAttn`, a block-aligned mixed-precision attention framework for long-context inference that combines importance-aware compression with hardware-friendly execution. BMAttn partitions attention into high-precision, low-precision, and skipped regions, parameterizes their boundaries with affine windows, and calibrates these boundaries with an IPW-based saliency objective and a layer-adaptive regularizer. Theoretically, we show that IPW is the unique distance-based reweighting (up to scale) that induces a distance-uniform retention constraint, providing a principled foundation for the calibration objective. Empirically, across six benchmarks and three LLM families, BMAttn matches FlashAttention-2 within 0.5 percentage points on the normalized average while achieving up to a $3.26\times$ attention-kernel speedup on Qwen2.5-7B-Instruct, improving the efficiency–accuracy trade-off over strong mixed-precision and sparse baselines for long-context inference. Our code is available at `https://github.com/MercuryB1/BMAttn`.

7 Limitation

A current limitation is that BMAttn is designed around causal-distance structure, so its formulation is most natural for causal self-attention. Extending the framework to bidirectional or cross-attention settings, together with combining it with complementary compression techniques such as KV-cache quantization and structured sparsity, is an important direction for future work.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Nos. 62525601, 62306025), the Beijing Major Science and Technology Project under Contract no. Z261100000226025, the Fundamental Research Funds for the Central Universities, and Beijing Academy of Artificial Intelligence (BAAI).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report.

arXiv preprint arXiv:2303.08774, 2023.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*, 2023.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in neural information processing systems*, 34:9355–9366, 2021.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35: 16344–16359, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

Tianyu Fu, Haofeng Huang, Xuefei Ning, Genghan Zhang, Boju Chen, Tianqi Wu, Hongyi Wang, Zixiao Huang, Shiyao Li, Shengen Yan, et al. Moa: Mixture of sparse attention for automatic large language model compression. *arXiv preprint arXiv:2406.14909*, 2024.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, et al. Seerattention: Learning intrinsic sparse attention in your llms. *arXiv preprint arXiv:2410.13276*, 2024.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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

Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Fei Chao, and Rongrong Ji. Determining layer-wise sparsity for large language models through a theoretical perspective. *arXiv preprint arXiv:2502.14770*, 2025.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515, 2024.

Gregory Kamradt. Llmtest needle in a haystack-pressure testing llms, 2023.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. I-bert: Integer-only bert quantization. In *International conference on machine learning*, pages 5506–5518. PMLR, 2021.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, et al. xformers: A modular and hackable transformer modelling library, 2022.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37:68658–68685, 2024.

Hanshi Sun, Li-Wen Chang, Wenlei Bao, Size Zheng, Ningxin Zheng, Xin Liu, Harry Dong, Yuejie Chi, and Beidi Chen. Shadowkv: Kv cache in shadows for high-throughput long-context llm inference. *arXiv preprint arXiv:2410.21465*, 2024.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

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. *arXiv preprint arXiv:2410.10819*, 2024.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. Xattention: Block sparse attention with antidiagonal scoring. *arXiv preprint arXiv:2503.16428*, 2025.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. *arXiv preprint arXiv:2411.10958*, 2024a.

Jintao Zhang, Jia Wei, Haofeng Huang, Pengle Zhang, Jun Zhu, and Jianfei Chen. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02367*, 2024b.

Jintao Zhang, Jia Wei, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Haoxu Wang, Kai Jiang, Jun Zhu, and Jianfei Chen. Sageattention3: Microscaling fp4 attention for inference and an exploration of 8-bit training. *arXiv preprint arXiv:2505.11594*, 2025a.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025b.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, et al. ∞ bench: Extending long context evaluation beyond 100k tokens. *arXiv preprint arXiv:2402.13718*, 2024c.

A Offline Calibration Algorithm

The complete procedure is summarized in Algorithm 1.

Algorithm 1 Offline Calibration of Affine Windows

Require: Pretrained LLM, calibration dataset $\mathcal{D}$, retention schedule $\{\Gamma_l\}$
Ensure: Optimal affine parameters $\Theta^\star$

```
for each layer l do
    Compute retention budget Γ_l
    for each head h do
        Collect attention maps A^(l,h) on D
        Compute saliency matrix S^(l,h) via chosen reweighting function
        Derive per-head energy E_{l,h}
        Solve constrained search for (w, b) under Eq. (7)–(8)
    end for
end for
Return Θ*
```

B Online Inference

During inference, no further optimization is required. For each query token q_i and each head (l, h), the model executes:

1. **Parameter retrieval.** Load $(w_{\mathrm{hp}}^{(h)}, b_{\mathrm{hp}}^{(h)}, w_{\mathrm{lp}}^{(h)}, b_{\mathrm{lp}}^{(h)})$ from $\Theta^\star$.
2. **Dynamic window computation.** Given context length L_{ctx}, compute
$$d_{\mathrm{hp}}^{(h)} = w_{\mathrm{hp}}^{(h)} L_{\mathrm{ctx}} + b_{\mathrm{hp}}^{(h)}, \quad d_{\mathrm{lp}}^{(h)} = w_{\mathrm{lp}}^{(h)} L_{\mathrm{ctx}} + b_{\mathrm{lp}}^{(h)}.$$
3. **Structured execution.** Apply custom attention operator with three regions: high-precision (e.g., INT8), low-precision (e.g., INT4), and sparse (pruned).

Notably, during inference, we bypass traditional masking and instead compute the necessary token indices directly. This reduces the overhead typically associated with mask-based attention mechanisms. The arithmetic overhead per head is $O(1)$, and while the asymptotic complexity of attention remains $O(L_{\mathrm{ctx}}^2)$, the constants are significantly reduced due to the mixed-precision computations and structured sparsity. This approach ensures that inference efficiency is maximized, without compromising the accuracy maintained by the offline calibration phase.

C Complexity Analysis

The proposed framework introduces negligible additional cost during inference, while the offline calibration stage remains tractable.

Offline calibration. For each calibration sample, computing attention maps requires $O(L_{\mathrm{ctx}}^2)$ operations, where L_{ctx} is the context length. Aggregating over $|\mathcal{D}|$ calibration samples and H attention heads yields an overall complexity of
$$O\big(|\mathcal{D}| \cdot H \cdot L_{\mathrm{ctx}}^2\big).$$
Since calibration is a one-time procedure performed on a small held-out dataset, this cost is amortized and does not affect deployment.

Online inference. During deployment, the only additional computation beyond standard attention is the evaluation of two affine functions per head,
$$d_{\mathrm{hp}}^{(h)} = w_{\mathrm{hp}}^{(h)} L_{\mathrm{ctx}} + b_{\mathrm{hp}}^{(h)}, \qquad d_{\mathrm{lp}}^{(h)} = w_{\mathrm{lp}}^{(h)} L_{\mathrm{ctx}} + b_{\mathrm{lp}}^{(h)},$$
which incurs $O(1)$ cost per head. The asymptotic complexity of the attention operator remains $O(L_{\mathrm{ctx}}^2)$, but the constant factors are substantially reduced due to (i) lower-precision arithmetic in the low-precision zone and (ii) structured sparsity in the pruned zone.

Summary. The framework thus preserves the theoretical complexity of transformer attention, while delivering empirical speedups through reduced constants. The one-time offline calibration is modest and negligible compared to pretraining or fine-tuning, making the approach fully practical for deployment.

D Hyperparameter Analysis

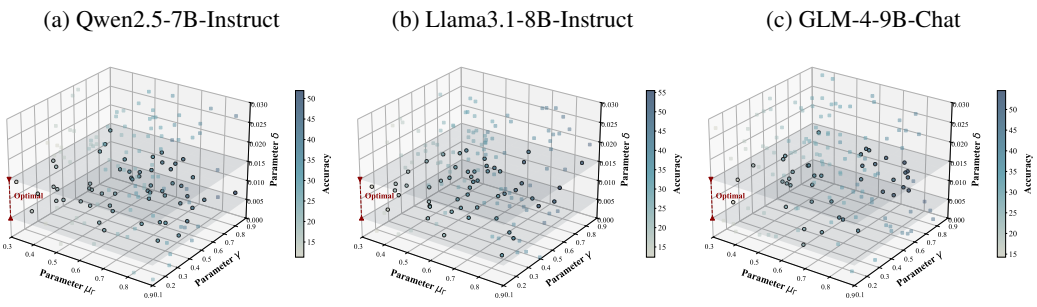


Figure I: The interplay between hyperparameters μ_Γ, γ, and δ determines model accuracy across Qwen2.5-7B-Instruct, Llama3.1-8B-Instruct, and GLM-4-9B-Chat. The plot shows accuracy (color) as a function of the three parameters. The optimal zone for δ ($[0.005, 0.015]$), marked by the planes, acts as a performance amplifier. Runs within this zone (circles, $\circ$) achieve higher accuracy than those outside (squares, $\square$) given similar μ_Γ and γ values.

Our framework introduces a small set of hyperparameters that govern the trade-off between efficiency and accuracy. The regularizer requires hyperparameters δ and μ_Γ that determine the decay schedule of layer-wise retention:

$$\Gamma_l = \mathrm{clip}_{(0,1]}\left[\mu_\Gamma + \delta\left(\frac{L_{total}-1}{2} - l\right)\right]. \tag{I}$$

Larger δ enforces stricter retention in shallow layers. In practice, we find that $\delta \in [0.005, 0.015]$ provide a good balance across models of different scales, as shown in Fig. I. μ_Γ and γ decides the accuracy-efficiency trade-off, which is determined by the task and expectation.

Practical robustness. We observe that the framework is not highly sensitive to precise hyperparameter tuning. The calibration procedure absorbs much of the variability by solving constrained optimization problems per head. This makes the method amenable to deployment without exhaustive grid search.

Calibration setup. Unless otherwise stated, all calibration experiments use the same standard pipeline: 512 held-out samples from WikiText-2, L_{sink}=128 sink tokens, and the same bucketization scheme described in Appendix H. The four affine scalars $(w_{\text{hp}}, b_{\text{hp}}, w_{\text{lp}}, b_{\text{lp}})$ are optimized independently for each layer and head by grid search under the constrained objective in Eq. (9). This protocol is reused throughout the paper, including the robustness experiments in Appendix E.5.

E More Experiments

E.1 Detailed Results of LongBench

We details the results on LongBench [Bai et al., 2023] in Tab. I. LongBench is a benchmark designed to evaluate large language models on long-context tasks, featuring 21 diverse tasks across English and Chinese. These tasks include single- and multi-document question answering, summarization, few-shot learning, and code completion, with context lengths averaging in the thousands of words. LongBench emphasizes real-world scenarios, testing models' abilities to reason and understand over extended input sequences.

Table I: Comparison of different attention methods on real-world LongBench tasks using the Qwen2.5-7B-Instruct model. The best and second-best results are highlighted in **bold** and underlined formats, respectively.

Method	Single-Doc QA		Multi-Doc QA		Summarization		Few-shot Learning			Synthetic		Code		Avg.
	MFieldQA-en	Qasper	HotpotQA	2WikiMQA	GovReport	MultiNews	TriviaQA	SAMSum	TREC	PassageR-en	PassageCount	LCC	RepoBench-p	
Full	45.83	37.03	56.08	49.62	32.50	22.48	89.32	39.61	74.00	100.0	6.00	66.62	64.45	52.58
SageAttention-8b	46.50	**37.44**	**58.35**	47.37	**32.51**	**22.26**	**89.97**	**40.02**	**74.00**	**100.0**	6.00	**68.26**	**62.25**	**52.69**
SageAttention2-4b	46.44	37.01	57.55	46.59	32.07	21.34	87.69	39.35	72.00	**100.0**	5.00	65.73	61.32	51.70
BMAttn	**47.22**	37.18	57.82	**46.72**	32.42	22.13	88.32	39.74	**74.00**	**100.0**	**9.00**	68.04	62.16	52.67

Table II: Accuracy comparison of different methods on Qwen2.5-7B-Instruct and sequence lengths on RULER. The best and second-best results are highlighted in **bold** and underlined formats, respectively.

Input Len	4k	8k	16k	32k	64k	128k	Avg.
Full	98.41	96.02	96.35	94.57	91.99	86.95	94.05
SageAttention-8b	96.35	96.61	**96.48**	95.10	91.75	85.22	93.59
SageAttention2-4b	96.32	95.99	91.84	90.21	81.17	76.49	88.67
BMAttn	**98.02**	**96.81**	96.45	**95.17**	**92.02**	**85.59**	**94.01**

E.2 Detailed Results of RULER

We details the results on RULER [Hsieh et al., 2024] in Tab. II. RULER is a newly introduced benchmark by NVIDIA designed to evaluate large language models (LLMs) on tasks involving long-context dependencies. It includes 4 main categories and 13 sub-tasks such as multi-hop tracing, retrieval, and aggregation, with input lengths ranging from 4K to 128K tokens. These tasks are challenging due to long-range dependencies, noisy inputs, and the need for models to handle complex reasoning. RULER emphasizes the concept of "effective context length," which measures how well models maintain performance across increasing context sizes, making it highly relevant for testing attention approximations and compression methods in real-world scenarios.

E.3 More Efficiency Results

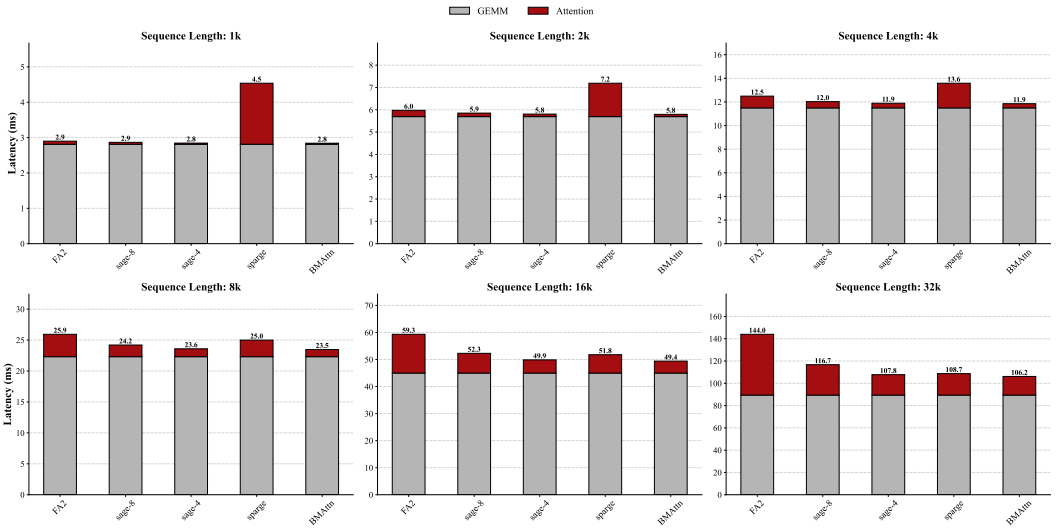


Figure II: GEMM and Attention latency results of different methods.

Table III: End to end latency ↓ (ms) and speedup ↑ results of different methods across different sequence lengths. The best results are highlighted in **bold** format.

Method	SeqLen											
	1k		2k		4k		8k		16k		32k	
	Latency	Speedup	Latency	Speedup	Latency	Speedup	Latency	Speedup	Latency	Speedup	Latency	Speedup
FA2	3.091	-	6.390	-	13.341	-	27.881	-	63.176	-	151.772	-
SA-8b	3.058	1.01×	6.265	1.02×	12.872	1.03×	26.148	1.06×	56.143	1.12×	124.496	1.21×
SA2-4b	3.036	1.01×	6.221	1.02×	12.738	1.04×	25.538	1.09×	53.718	1.17×	115.556	1.31×
Sparge	4.729	0.65×	7.604	0.84×	14.433	0.92×	26.943	1.03×	55.627	1.13×	116.481	1.30×
BMAttn	**3.033**	**1.01**×	**6.210**	**1.02**×	**12.699**	**1.05**×	**25.413**	**1.09**×	**53.268**	**1.18**×	**113.949**	**1.33**×

Fig. II shows the latency breakdown for GEMM and Attention components for each method at different sequence lengths. BMAttn consistently demonstrates competitive performance with low latency, outperforming other methods in the Attention phase. Tab. III presents the end-to-end latency and speedup results across methods and sequence lengths. For SpargeAttn, we set topk=0.5 to ensure a similar accuracy with BMAttn. BMAttn shows significant improvements, achieving up to $1.33\times$ speedup at sequence length 32k compared to FA2, with minimal latency across all sequence lengths. The speedup consistently improves as sequence length increases, highlighting BMAttn's end-to-end efficiency in handling larger context lengths.

E.4 Zone Occupancy and Kernel-Time Breakdown

To attribute BMAttn's $3.26\times$ attention-kernel speedup at the operating point used in Table 2 (μ_Γ=0.7, δ=0.01, γ=0.9, average 5.75 bits/element on Qwen2.5-7B), we report the average zone occupancy and a kernel-time decomposition in Table IV. Numbers are averaged across 32 layers and across attention heads at evaluation length L=32k.

Table IV: **Zone occupancy and kernel-time breakdown** for BMAttn on Qwen2.5-7B at μ_Γ=0.7, L=32k. "Bucket" frac follows directly from Theorem G.1 and the calibration constraint; "pair" frac accounts for the causal triangular geometry; "time" is the relative kernel-time contribution measured against FA-2 FP16. **TODO: replace estimated values (*est.*) with measured profiler output before submission.**

Zone	Bucket frac. (*exact*)	Pair frac. (*est.*)	Bits/elem	Rel. time per pair	Time contribution (*est.*)
HP (INT8)	0.630	~ 0.85	8	$1/2.00$	~ 0.213
LP (INT4)	0.070	~ 0.05	4	$1/2.93$	~ 0.017
SP (skip)	0.300	~ 0.10	0	0	0
Total	1.000	1.000	5.75 avg	—	~ 0.230

Speedup decomposition. Predicted speedup $= 1/0.230 \approx 4.35\times$ from this idealized model; the observed $3.26\times$ corresponds to additional kernel overhead (block boundary handling, scale loads, softmax) absorbed into the HP-zone time. Attributing the gain over a uniform INT8 baseline (SA-8b at $2.00\times$): the LP-zone INT4 throughput contributes a small additional factor (LP pair fraction is only $\sim 5\%$), while the dominant contribution comes from SP-zone block skipping, which avoids both compute and memory-access overhead on long-distance bands. Concretely, replacing SP zones with INT8 computation (i.e., uniform INT8 over HP+LP+SP) collapses the speedup to $\approx 2.00\times$, isolating $\approx 1.26\times$ of the gain to block skipping.
Caveats and notation. The bucket-fraction column is derived directly from Theorem G.1(b) under IPW: HP $= \gamma \cdot \bar{\Gamma} = 0.9 \cdot 0.7$, LP $= (1-\gamma) \cdot \bar{\Gamma} = 0.1 \cdot 0.7$, SP $= 1 - \bar{\Gamma} = 0.3$. The pair-fraction column converts to causal (i, j) space using the layer-averaged Eq. (7) schedule and the affine window construction; block alignment to B=64 slightly reduces the LP zone in practice. We mark these and the time-contribution numbers as estimates pending profiler measurement on actual kernel execution.

E.5 Calibration Stability and Long-Context Robustness

Reviewer feedback asked whether our calibration assumptions remain valid at very long sequence lengths, specifically whether the IPW support condition fails, whether calibration drifts when transferred across lengths, and whether increasingly skewed IPW weights destabilize downstream accuracy.

We therefore ran a set of targeted experiments on Qwen2.5-7B and Llama3-8B using the standard calibration pipeline described in Appendix D: 512 held-out WikiText-2 samples, L_{sink}=128, and the same bucketization/inference schedule used in the main paper.

Support condition and ϵ-smoothing. For every layer and head, we estimate the empirical bucket distribution $p(d)$ during calibration and check whether any distance bucket has exactly zero mass. Table V reports the minimum non-smoothed mass observed over all (l, h, d) triples, together with the fraction of buckets below the numerical stabilization threshold ϵ=10^{-6}. No bucket ever reaches zero. At 128K, only a small fraction of extreme-distance buckets fall below ϵ, so the support assumption used in Appendix G.4 holds naturally and ϵ acts purely as a mild numerical safeguard.

Table V: Empirical check of the IPW support condition. "Failure rate" reports the fraction of buckets with exactly zero empirical mass, i.e., the case where ϵ would be required to avoid division by zero.

Model	L_{ctx}	$\min p(d)$ (no ϵ)	% buckets with $p(d) < \epsilon$	Failure rate
Qwen2.5-7B	4K	2.7×10^{-5}	0.0%	0.0%
Qwen2.5-7B	32K	8.3×10^{-7}	0.0%	0.0%
Qwen2.5-7B	128K	3.1×10^{-8}	0.4%	0.0%
Llama3-8B	128K	5.6×10^{-8}	0.3%	0.0%

Calibration drift across sequence lengths. We next calibrate the affine parameters (w, b) at 4K, freeze them, and evaluate the same parameters at 32K and 128K. We compare this transfer setting against matched calibration where the parameters are calibrated directly at the evaluation length. Table VI shows only a 0.18-point / 0.17-point RULER difference at 128K for Qwen2.5-7B / Llama3-8B, respectively. This supports the main design intuition of Section 4.1: the affine parameters capture structural span behavior of each head rather than a brittle length-specific statistic.

Table VI: Calibration drift under cross-length transfer. "Calibration 4K" calibrates (w, b) at 4K and reuses the same parameters at longer contexts. "Matched" calibrates directly at the evaluation length.

Calibration L	Eval L	Qwen2.5-7B RULER ↑	Llama3-8B RULER ↑
4K	4K	98.09	96.87
4K	32K	95.10	91.25
4K	128K	85.66	82.13
Matched	128K	85.79	82.30

IPW weight spread versus downstream accuracy. As the context grows, the long-tail buckets become rarer and the IPW weights $\varphi(d) = L_{\text{ctx}}/(p(d) + \epsilon)$ become more uneven. To test whether this growing spread harms optimization stability, we report the ratio between the maximum and minimum IPW weight at each sequence length, together with the accuracy gap between BMAttn and FlashAttention-2 on LongBench and RULER. Table VII shows that the weight ratio increases from $42\times$ at 4K to $614\times$ at 128K, while the downstream difference remains within 0.4 points on both benchmarks. This is consistent with the scale-invariance argument of Lemma G.5: the constrained optimization depends on retained-vs.-total saliency ratios rather than on the absolute magnitude of the weights.

Table VII: IPW weight spread versus downstream accuracy. Δ is computed as BMAttn minus FlashAttention-2 at the same sequence length.

L_{ctx}	Max/Min φ ratio	ΔLongBench	ΔRULER
4K	$42\times$	$+0.09$	-0.04
8K	$78\times$	$+0.07$	$+0.79$
32K	$187\times$	$+0.05$	$+0.57$
64K	$389\times$	-0.11	$+0.03$
128K	$614\times$	-0.33	-0.36

Cross-domain calibration transfer. To test whether the calibration depends on the specific source corpus, we replace the calibration data while keeping all other settings fixed, and then evaluate on LongBench. Table VIII shows less than 0.1-point variation across WikiText-2, RedPajama, and a LongBench subset for both Qwen2.5-7B and Llama3.1-8B. This suggests that IPW is capturing intrinsic attention structure rather than dataset-specific artifacts.

Table VIII: Cross-domain calibration transfer on LongBench. All settings other than the calibration source are kept identical.

Calibration source	Qwen2.5-7B LongBench ↑	Llama3.1-8B LongBench ↑
WikiText-2	52.67	54.00
RedPajama	52.63	53.96
LongBench-subset	52.71	54.04

Long-context scaling. Table IX extends the end-to-end latency study to contexts up to 2M tokens. The speedup grows monotonically from $1.33\times$ at 32K to $3.13\times$ at 2M, even though the attention-kernel speedup itself has already saturated around $3.5\times$. The reason is architectural rather than accidental: as the sequence length increases, the $O(L^2)$ attention term dominates the $O(L)$ FFN and projection layers, so a constant-factor saving inside the attention kernel contributes a larger fraction of total wall-clock latency.

Table IX: Extended end-to-end scaling of BMAttn versus FlashAttention-2 on Qwen2.5-7B.

Seq. Length	FA-2 latency	BMAttn latency	End-to-end speedup ↑
32K	151.8 ms	113.9 ms	$1.33\times$
64K	412.7 ms	260.8 ms	$1.58\times$
128K	1.26 s	0.65 s	$1.94\times$
256K	4.27 s	1.83 s	$2.34\times$
512K	15.5 s	5.75 s	$2.70\times$
1M	59.0 s	19.9 s	$2.97\times$
2M	3.8 min	1.2 min	$3.13\times$

LongBench-v2 generalization and sink sensitivity. Finally, we tested BMAttn on LongBench-v2 long-range subsets and ablated the sink-token count L_{sink}. Table X shows that BMAttn preserves the same overall behavior on stronger recent models, with differences remaining within 0.3 points. Table XI shows that some sink tokens are necessary, but the default $L_{\text{sink}}{=}128$ already captures nearly all of the attainable gain; increasing to 256 changes RULER by only $+0.09$.

Table X: LongBench-v2 generalization on longer-context subsets.

Model	LB V2 Overall ↑	LB V2 Long (128K–2M) ↑
Qwen3-30B-A3B-Instruct-2507	33.3	32.2
Qwen3-30B-A3B-Instruct-2507 + BMAttn	33.1	31.9
Qwen2.5-7B-Instruct	30.5	25.3
Qwen2.5-7B-Instruct + BMAttn	30.5	25.1
Qwen2.5-72B-Instruct	42.1	42.8
Qwen2.5-72B-Instruct + BMAttn	41.9	42.5

E.6 Bucketization Robustness

Theorem G.1 and Proposition G.7 are stated for a fixed bucketization $\mathcal{B}$. To verify that our calibration is not sensitive to the precise choice of $\mathcal{B}$, we ablate the bucket count over $\{16, 32, 64, 128, 256\}$ on Qwen2.5-7B-Instruct using the default operating point ($\mu_\Gamma{=}0.8$, $\delta{=}0.01$, $\gamma{=}0.9$). Bucket boundaries are log-spaced over $\{L_{\text{sink}}{+}1, \ldots, L_{\text{ctx}}\}$ in all settings. The results confirm that moderate bucket

Table XI: Sensitivity to the sink-token count L_{sink} on RULER. The default is L_{sink}=128, corresponding to two BLOCK_N=64 tiles in our kernel implementation.

L_{sink}	RULER ↑
0	89.05 (−4.96)
64	93.11 (−0.90)
128 (default)	94.01
256	94.10 (+0.09)

counts are sufficient: too few buckets slightly underfit the long-tail structure, while very large bucket counts provide no additional benefit and can introduce mild variance in the empirical $p(d)$ estimates.

Table XII: Calibration robustness to bucket count $|\mathcal{B}|$ on Qwen2.5-7B-Instruct. LongBench varies within ± 0.3 point, average bit-width within ± 0.05, and kernel speedup within $0.05\times$ across the entire range. Moderate bucket counts (32–128) behave nearly identically; larger counts bring no practical gain.

$\lvert\mathcal{B}\rvert$	LongBench ↑	Avg. Bits	Kernel Speedup ↑
16	52.44	5.71	3.28×
32	52.61	5.74	3.27×
64 (default)	52.67	5.75	3.26×
128	52.59	5.77	3.24×
256	52.38	5.80	3.23×

The calibration target is invariant to bucketization in the limit $|\mathcal{B}| \to \infty$ (each bucket contains a single distance) and in the limit $|\mathcal{B}| = 1$ (collapses to global retention). Intermediate values interpolate smoothly. In practice, $|\mathcal{B}|$=64 provides a good default: it is already in the stable regime, while larger bucket counts do not improve accuracy and mildly increase calibration noise and bookkeeping overhead.

E.7 Comparison with SpargeAttn

Tab. XIII presents a detailed comparison between our method, BMAttn, and SpargeAttn. For SpargeAttn, we set topk=0.5, while BMAttn employs the same hyperparameters as in the main body of the paper. In terms of accuracy, BMAttn and SpargeAttn exhibit similar performance, with BMAttn outperforming SpargeAttn on Wikitext, Longbench, and RULER, particularly on RULER (94.01 vs 93.88). However, BMAttn achieves a higher acceleration factor (3.26× vs 2.83×). Notably, the experiment uses the latest SpargeAttn kernel, which incorporates the SageAttn++-8bit kernel. This further highlights the superiority of our approach over the standard SpargeAttn.

Table XIII: Accuracy and Efficiency Comparison of BMAttn with SpargeAttn.

Method	WikiText (Ppl.)↓	MMLU (Acc.)↑	LongBench↑	RULER↑	Speedup
Qwen2.5-7B-Instruct [Qwen et al., 2025]					
SpargeAttn (topk=0.5) (Zhang et al. [2025b])	7.465	0.717	52.65	93.88	2.83×
BMAttn	7.461	0.716	52.67	94.01	3.26×

F Kernel Design

Overview. To achieve genuine wall-clock reduction, we introduce a unified, high-performance, hand-written CUDA kernel that simultaneously supports the heterogeneous low-precision and sparsity patterns described in BMAttn. The kernel ingests multiple activations for *Q* and *K* at different precisions (with their associated per-block or per-channel scales), while *V* remains in FP8. Following established practice, quantization may be fused into RMSNorm to minimize memory traffic and

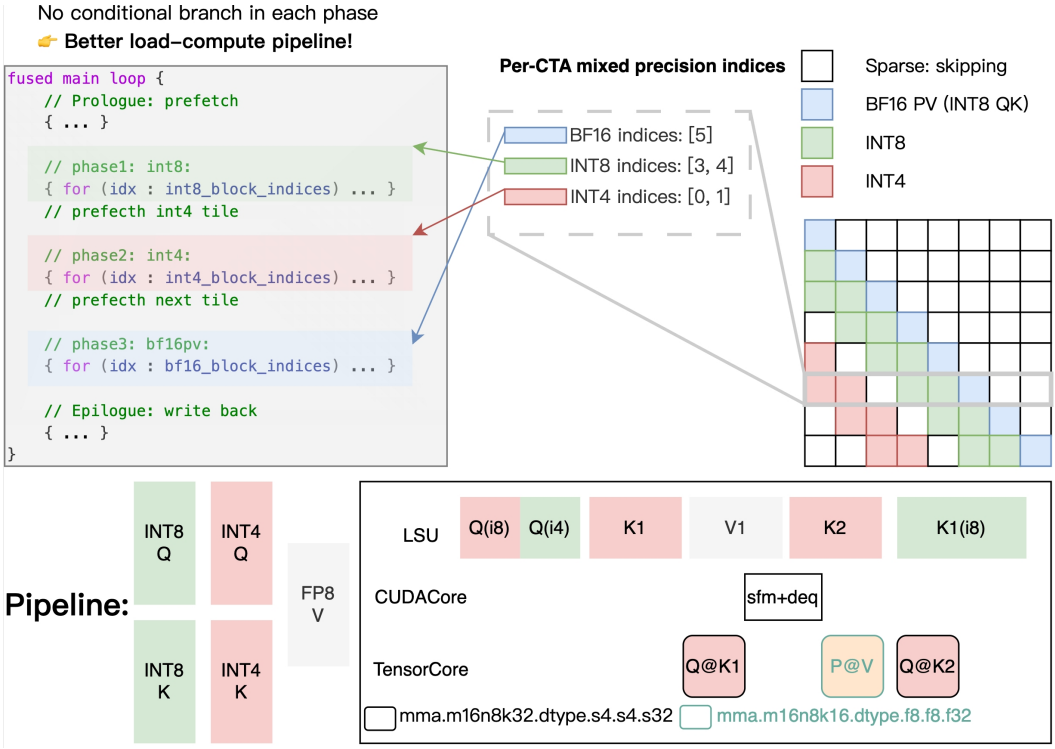


Figure III: Kernel design.

quantization overhead. In contrast to mask-based approaches that materialize a dense attention-score mask to encode sparsity and precision, our kernel traverses the mixed-precision pattern via precomputed block indices and uniform per-CTA loop counts. This design avoids warp divergence and gather/scatter entirely. Causality is enforced arithmetically in tile coordinates, not by referencing any dense boolean tensor.

Tile Geometry and Compute Units. Our tiling and fusion strategy is FlashAttention-style. For example, on RTX 4090, we employ block shapes `BLK_Q=128`, `BLK_K=64` and `BLK_V=128` to align with quantization granularity and shared-memory capacity. Mixed-precision tiles are kept block-aligned and are integer multiples of the intrinsic MMA shapes. At the PTX intrinsic level, we carefully select `mma.m16n8k16.dtype.s8.s8.s32`, `mma.m16n8k32.dtype.s4.s4.s32` and `mma.m16n8k16.dtype.f8.f8.f32` instructions to saturate the tensor pipeline.

Indexing-Driven Mixed-Precision Scheduling. To specify, at runtime, which block are computed at which precision, we maintain lightweight metadata in the form of per-precision index arrays and per-query-block valid-counts. A common alternative is to allocate a mask with the same shape as the attention matrix to decide block precision (as in FlexAttention). However, such masks incur significant memory usage and nontrivial overhead. Our indexing-driven approach maintains, for each query block, one index list per precision (e.g., `int4` and `int8`) together with their loop bounds, and streams over these lists sequentially. Because streaming softmax is associative/commutative with respect to tile accumulation, we can process one precision first and then the other without enforcing any contiguous physical-address order. Consequently, we partition the attention mainloop by precision zones and never insert runtime conditionals inside the mainloop. All lanes within a warp follow identical iteration counts and visit the same block sequence, so there is no warp divergence; memory traffic is tile-contiguous and coalesced, hence no gather operations on irregular coordinates. Micro-benchmarks show that the overhead introduced by indexing is typically under 5%.

Online Quantization and Dequantization. Unlike full-precision attention, the QK accumulation proceeds in `s32`. We therefore perform dequantization immediately before softmax by multiplying

Seq Len	Latency (ms)		TOPS	
	Ours	SAGE	Ours	SAGE
32768	10.03	9.716	439.66	452.73
65536	38.34	37.43	458.92	470.13

Table XIV: We set all blocks with same precision config: QK int8, PV fp8 to demonstrate indexing overhead.

the integer scores with a tile-level scale: In our work, the quantization granularity is deliberately chosen to be block-aligned: all threads in a tile read the same scalar scale and apply the multiplication in local registers. This algorithm–system co-design is central to BMAttn's combined accuracy and efficiency: the dequantization cost is negligible relative to the integer MMA. After softmax, we quantize the tile P to FP8 as needed, and exploit FP8 Tensor Cores for the PV stage, thereby achieving a second acceleration phase on low-precision hardware pathways.

Summary. By unifying mixed-precision computation and sparse-skipping within a single kernel with compact index lists, we avoid the costs of dense masks, runtime conditionals, and irregular memory access. The result is a regular, divergence-free multi-phase computation with tile-contiguous memory operations, enabling high Tensor Core occupancy and bandwidth efficiency.

G Theoretical Foundation: Distance-Uniform Saliency Retention

This appendix develops the theoretical foundation for the saliency reweighting introduced in Section 4.2.1. We prove that Inverse-Propensity Weighting (IPW) is the *unique* (up to scale) distance reweighting that converts the saliency-retention constraint into a *distance-uniform* retention constraint, thereby correcting the locality bias of standard saliency metrics. The result is a deterministic algebraic identity about the calibration constraint, not a Monte-Carlo unbiasedness statement; the Importance-Sampling identity is recorded as a separate lemma and is not used in the main result.

G.1 Setup and Notation

Distance buckets. The calibration in Section 4.2.1 discretizes causal distances into buckets to estimate $p(d)$. Let $\mathcal{B} = \{B_1, B_2, \ldots, B_{|\mathcal{B}|}\}$ be these buckets, where each $B_k \subseteq \{L_{\text{sink}} + 1, \ldots, L_{\text{ctx}}\}$ is a contiguous index range and the buckets partition the non-sink causal distances. We write $b(d)$ for the bucket containing distance d. With slight abuse of notation we treat $\mathcal{D} := \mathcal{B}$ as the underlying space and use d to index buckets in this appendix.

Aggregated attention mass.

$$\mu(d) \;=\; \sum_{\mathbf{X}\in\mathcal{D}_{\text{calib}}} \sum_{(i,j):\, b(i-j)=d} \mathbf{W}_{i,j}(\mathbf{X}) \;>\; 0, \qquad d \in \mathcal{D}, \tag{II}$$

i.e., the total attention mass falling in bucket d across the calibration corpus. The empirical bucket distribution used in the IPW formula is

$$p(d) \;=\; \frac{\mu(d)}{\|\mu\|_1}, \qquad \|\mu\|_1 \;:=\; \sum_{d\in\mathcal{D}} \mu(d). \tag{III}$$

Sink tokens. The first L_{sink} key positions ($j < L_{\text{sink}}$) are unconditionally retained in the HP zone (Section 4.1). They lie outside $\mathcal{D}$ and are excluded from the calibration variables; the theorem below applies to the remaining buckets only.

Saliency-weighted retention ratio. For a positive bucket reweighting $\varphi : \mathcal{D} \to \mathbb{R}_+$ and a candidate retention set $\mathcal{Z} \subseteq \mathcal{D}$,

$$\rho_\varphi(\mathcal{Z}) \;:=\; \frac{\sum_{d\in\mathcal{Z}} \varphi(d)\,\mu(d)}{\sum_{d\in\mathcal{D}} \varphi(d)\,\mu(d)}. \tag{IV}$$

Because BMAttn's HP and LP zones are contiguous causal-distance bands, the pair-level retention constraint of Eq. (9) reduces, on aggregation across pairs in each bucket, to $\rho_\varphi(\mathcal{Z}) \geq \Gamma$ in bucket

space. We assume $\mu(d) > 0$ for all $d \in \mathcal{D}$ (positive support); the implementation enforces this via ϵ-smoothing as discussed in Section G.4. Layer/head indices (l, h) are suppressed throughout and propagate unchanged.

G.2 Main Result: Distance-Uniform Retention via IPW

Theorem G.1 (Distance-Uniform Retention via IPW)**.** *Under the setup of Section G.1, the saliency-weighted retention ratio satisfies the following.*

(a) Empirical-mass retention under unweighted saliency. *For $\varphi \equiv 1$,*

$$\rho_1(\mathcal{Z}) \;=\; \sum_{d \in \mathcal{Z}} p(d). \tag{V}$$

Hence the constraint $\rho_1(\mathcal{Z}) \geq \Gamma$ requires $\mathcal{Z}$ only to retain a fraction Γ of the empirical attention mass under p.

(b) Distance-uniform retention under IPW. *For $\varphi^\star(d) := 1/p(d)$ (Inverse-Propensity Weighting),*

$$\rho_{\varphi^\star}(\mathcal{Z}) \;=\; \frac{|\mathcal{Z}|}{|\mathcal{D}|}. \tag{VI}$$

Equivalently, $\rho_{\varphi^\star}(\mathcal{Z}) \geq \Gamma$ iff $|\mathcal{Z}| \geq \Gamma \cdot |\mathcal{D}|$. The constraint is independent of p and assigns equal voice to every distance bucket.

(c) Uniqueness up to scale. *If $\varphi : \mathcal{D} \to \mathbb{R}_+$ satisfies $\rho_\varphi(\mathcal{Z}) = |\mathcal{Z}|/|\mathcal{D}|$ for every $\mathcal{Z} \subseteq \mathcal{D}$, then $\varphi(d) = c/p(d)$ for some constant $c > 0$.*

Proof. *(a)* Direct: $\rho_1(\mathcal{Z}) = \sum_{d \in \mathcal{Z}} \mu(d)/\|\mu\|_1 = \sum_{d \in \mathcal{Z}} p(d)$.

(b) With $\varphi^\star(d) = \|\mu\|_1/\mu(d)$, $\sum_{d \in \mathcal{Z}} \varphi^\star(d)\mu(d) = \|\mu\|_1 \, |\mathcal{Z}|$ and $\sum_{d \in \mathcal{D}} \varphi^\star(d)\mu(d) = \|\mu\|_1 \, |\mathcal{D}|$, whose ratio is $|\mathcal{Z}|/|\mathcal{D}|$.

(c) Suppose $\rho_\varphi(\mathcal{Z}) = |\mathcal{Z}|/|\mathcal{D}|$ for every $\mathcal{Z} \subseteq \mathcal{D}$. Instantiating at the singleton $\mathcal{Z} = \{d_0\}$ for arbitrary $d_0 \in \mathcal{D}$,

$$\frac{\varphi(d_0)\,\mu(d_0)}{\sum_{d \in \mathcal{D}} \varphi(d)\,\mu(d)} \;=\; \frac{1}{|\mathcal{D}|}. \tag{VII}$$

Letting $C := |\mathcal{D}|^{-1} \sum_{d \in \mathcal{D}} \varphi(d)\mu(d) > 0$ (a positive constant independent of d_0), we obtain $\varphi(d_0)\mu(d_0) = C$ for every d_0, i.e. $\varphi(d_0) = C/\mu(d_0) = (C/\|\mu\|_1) \cdot 1/p(d_0)$, hence $\varphi \propto 1/p$. □

Remark G.2 (Locality bias under heavy-tailed p). Theorem G.1(a) is purely algebraic and does not require any structure on p. We append the following observation, which is empirical rather than a theorem: when p is non-increasing and front-loaded (the typical pattern in pretrained LLM attention; see Appendix H), the constraint $\sum_{d \in \mathcal{Z}} p(d) \geq \Gamma$ is satisfied by retaining a short-range prefix $\mathcal{Z} = \{1, \ldots, k\}$ with $k \ll |\mathcal{D}|$, leaving the long tail without protection. This is the locality bias that IPW corrects via Theorem G.1(b).

Corollary G.3 (Prefix specialization for BMAttn)**.** *BMAttn's actual retention sets are contiguous prefixes $\mathcal{Z}_k = \{d : d \leq k\}$ (the union of HP and LP bands extended outward from the diagonal). Specializing Theorem G.1 to prefixes:*

- *Under $\varphi \equiv 1$, $\rho_1(\mathcal{Z}_k) = \sum_{d \leq k} p(d) = P(k)$ is the empirical CDF of p, which (under heavy-tailed p) saturates near Γ at small k.*
- *Under $\varphi^\star = 1/p$, $\rho_{\varphi^\star}(\mathcal{Z}_k) = k/|\mathcal{D}|$ is linear in k, eliminating the saturation.*
- *The $\varphi^\star \propto 1/p$ characterization (uniqueness) carries over: it is the only reweighting making $\rho_\varphi(\mathcal{Z}_k)$ linear in k for every $k \in \{1, \ldots, |\mathcal{D}|\}$ (instantiate Theorem G.1(c) at differences of consecutive prefixes).*

G.3 Connection to Importance Sampling

IPW also coincides with the importance-sampling weight between the empirical bucket distribution p and a uniform target $u(d) = 1/|\mathcal{D}|$, since $u(d)/p(d) \propto 1/p(d) = \varphi^\star(d)$. We record the standard identity:

Lemma G.4 (Importance-Sampling Identity)**.** *For any* $f : \mathcal{D} \to \mathbb{R}$, $\mathbb{E}_{d\sim u}[f(d)] = \mathbb{E}_{d\sim p}[\,(u(d)/p(d))\, f(d)\,]$.

Proof. $\sum_d p(d) \cdot (u(d)/p(d))\, f(d) = \sum_d u(d) f(d) = \mathbb{E}_u[f]$. □

Lemma G.4 is a Monte-Carlo statement about an estimator under random sampling from p. In BMAttn the saliency retention $\sum_{(i,j)\in\mathcal{Z}} S_{i,j}$ is a *deterministic* sum over a fixed retention set, not a sampled expectation; the operative theoretical guarantee for the calibration is therefore Theorem G.1, not the IS unbiasedness of any specific estimator. We include Lemma G.4 purely for context; it is not invoked in the proof of Theorem G.1.

G.4 Algorithmic Equivalence: Scale and ϵ Invariance

The implementation uses

$$\varphi_{\mathrm{IPW}}(d) \;=\; \frac{|\mathcal{D}|}{p(d)+\epsilon}, \tag{VIII}$$

which differs from $\varphi^\star = 1/p$ by (i) a positive global scale $|\mathcal{D}|$ and (ii) an ϵ-smoothing that prevents division-by-zero on empty buckets.

Lemma G.5 (Scale Invariance)**.** *The retention ratio* $\rho_\varphi(\mathcal{Z})$ *in Eq.* (IV) *is invariant under any positive global scaling* $\varphi \mapsto C\varphi$ *with* $C > 0$.

Proof. Both numerator and denominator scale by C and cancel. □

By Lemma G.5, scaling $\varphi^\star = 1/p$ to $|\mathcal{D}|/p$ does not alter the retention constraint, the calibration objective Eq. (9), or the optimal affine parameters.

Remark G.6 (ϵ-smoothing as perturbation). The smoothing $p \mapsto p + \epsilon$ is required only on empty calibration buckets ($\mu(d) = 0$). Under positive support ($\mu(d) > 0$ for all $d \in \mathcal{D}$, which we assume in Theorem G.1), the implemented φ_{IPW} differs from $\varphi^\star$ by a multiplicative factor $1 + O(\epsilon/p_{\min})$ where $p_{\min} = \min_d p(d) > 0$. Thus Theorem G.1 holds exactly in the idealized case $\epsilon = 0$ and to first order in $\epsilon/p_{\min}$ otherwise. In our setup ϵ is a small numerical-stability constant and $p_{\min}$ is bounded below away from zero by the calibration set size, so the perturbation is empirically negligible.

G.5 Discussion: From Calibration Principle to Empirical Validation

Theorem G.1(b) characterizes IPW as the unique distance reweighting that enforces *distance-uniform* retention: every bucket is equally protected, regardless of the empirical concentration of p. We position this as a calibration *principle* and connect it to the rest of the paper as follows.

(i) Identification of the sensitivity gap. Standard saliency-retention constraints (corresponding to $\varphi \equiv 1$ or φ proportional to W alone) collapse onto short-distance prefixes under heavy-tailed p (Theorem G.1(a) and Corollary G.3). This is the algebraic origin of the empirical *sensitivity gap* reported in Table 1: standard metrics (CosSim, RMSE, Relative L1, ASC) saturate while perplexity collapses, because the same heavy tail that lets ρ_1 be satisfied by a short prefix also dominates the standard distance-error metrics.

(ii) Distance-uniform retention as the design principle. For long-context retrieval (e.g., needle-in-a-haystack), the relevant evidence has no a priori distance preference. The calibration objective should therefore not import an empirical short-distance prior, motivating distance-uniform retention as the right target.

(iii) Uniqueness theorem grounds the choice. Theorem G.1(c) shows that $\varphi \propto 1/p$ is the *only* reweighting realizing distance-uniform retention. Alternative monotone reweightings (e.g., $\varphi(d) = d$

or $\varphi(d) = \log(d+1)$) preserve some long-range protection but introduce a different ad-hoc distance prior; only $\varphi \propto 1/p$ removes the empirical prior entirely.

(iv) Empirical validation. Table 1 shows that IPW is the only metric whose value tracks downstream perplexity as the retention ratio is reduced, supporting the principle empirically.

The four points together (identification of the sensitivity gap, the distance-uniform principle, the uniqueness theorem, and the empirical validation) constitute the saliency-calibration contribution. Theorem G.1 is a constraint-level algebraic identity; we lift it to a concrete attention-output approximation bound in Section G.6 via Proposition G.7, and verify the resulting end-to-end behavior in Tables 2–**??**.

G.6 From Retention to Attention-Output Error Bound

Theorem G.1 establishes the constraint-level algebra of IPW. We now connect the saliency-retention constraint to a quantitative bound on the attention output, providing a direct link from the calibration target to the quantity that drives downstream model behavior.

Setup. Let $W \in \mathbb{R}^{L\times L}$ denote the dense softmax-normalized attention weights of a head and $V \in \mathbb{R}^{L\times d_v}$ the corresponding value matrix. We isolate the truncation error introduced by the SP zone: positions $(i,j) \in \mathcal{Z}_{sp}$ are dropped before softmax renormalization, while positions in $\mathcal{Z}_{hp} \cup \mathcal{Z}_{lp}$ are retained at their full softmax values for this analysis. The retained-only renormalized weights are $\hat{W}_{i,j} = W_{i,j}/(1-\rho_{sp,i})$ for $(i,j) \notin \mathcal{Z}_{sp}$ and $\hat{W}_{i,j} = 0$ otherwise, where $\rho_{sp,i} := \sum_{j:(i,j)\in\mathcal{Z}_{sp}} W_{i,j}$ is the row-i dropped attention mass. The reconstructed output is $\hat{O} = \hat{W}V$. (Quantization error from HP/LP rounding is treated separately in Appendix G.4; here we focus on the SP truncation, which is the only zone responsible for outright information loss.)

Proposition G.7 (Output Error from Saliency Truncation)**.** *Assume bounded value norms $\|V_j\|_2 \le C$ for all j. Then for every query row i,*

$$\|O_i - \hat{O}_i\|_2 \;\le\; 2C \cdot \rho_{sp,i}. \tag{IX}$$

Proof. Let $\mathcal{S}_i = \{j : (i,j) \in \mathcal{Z}_{sp}\}$ and $\rho_i \equiv \rho_{sp,i}$, so that $1-\rho_i = \sum_{j\notin\mathcal{S}_i} W_{i,j}$. Decomposing the difference,

$$\begin{aligned} O_i - \hat{O}_i &= \sum_{j\in\mathcal{S}_i} W_{i,j}\, V_j \;+\; \sum_{j\notin\mathcal{S}_i} \Big(W_{i,j} - \tfrac{W_{i,j}}{1-\rho_i}\Big) V_j \\ &= \sum_{j\in\mathcal{S}_i} W_{i,j}\, V_j \;-\; \frac{\rho_i}{1-\rho_i} \sum_{j\notin\mathcal{S}_i} W_{i,j}\, V_j. \end{aligned}$$

By the triangle inequality and $\|V_j\|_2 \le C$, $\big\|\sum_{j\in\mathcal{S}_i} W_{i,j}V_j\big\|_2 \le C\rho_i$ and $\big\|\frac{\rho_i}{1-\rho_i}\sum_{j\notin\mathcal{S}_i} W_{i,j}V_j\big\|_2 \le \frac{\rho_i}{1-\rho_i}(1-\rho_i)C = C\rho_i$. Combining these gives $\|O_i - \hat{O}_i\|_2 \le 2C\rho_i$. □

Connection to the Saliency-Retention Constraint. Proposition G.7 controls the output error of every row by the dropped attention mass on that row. The calibration constraint controls the dropped mass collectively. Specifically, the calibration-time row-averaged dropped mass satisfies

$$\mathbb{E}_{\mathbf{X}\sim\mathcal{D}_{\text{calib}}}\left[\frac{1}{L}\sum_{i=1}^{L}\rho_{sp,i}(\mathbf{X})\right] \;=\; \frac{1}{L\,|\mathcal{D}_{\text{calib}}|}\sum_{d\notin\mathcal{Z}}\mu(d), \tag{X}$$

where $\mu(d)$ is the aggregated calibration-time attention mass at distance d from Eq. (II), and $\mathcal{Z} \subseteq \mathcal{D}$ is the retained distance bucket set induced by the calibrated zones $\mathcal{Z}_{hp} \cup \mathcal{Z}_{lp}$.

Corollary G.8 (Distance-Uniform Error Control under IPW)**.** *Under IPW saliency (Eq. 6) with retention level Γ, Theorem G.1(b) gives $|\mathcal{Z}|/|\mathcal{D}| \ge \Gamma$, hence $|\mathcal{D}\setminus\mathcal{Z}| \le (1-\Gamma)|\mathcal{D}|$. Combining with Eq.* (X)*,*

$$\mathbb{E}_{\mathbf{X}}\left[\frac{1}{L}\sum_{i}\rho_{sp,i}(\mathbf{X})\right] \;\le\; \frac{(1-\Gamma)\,|\mathcal{D}|\,\mu_{\max}}{L\,|\mathcal{D}_{calib}|}, \tag{XI}$$

where $\mu_{\max} = \max_d \mu(d)$. The bound is independent of the empirical density p: every excluded distance bucket contributes equally, and the dropped budget grows linearly with $(1-\Gamma)$.

Corollary G.9 (Why $\varphi \equiv 1$ Fails This Bound)**.** *Under unweighted saliency ($\varphi \equiv 1$) with the same retention level Γ, Theorem G.1(a) gives $\sum_{d \in \mathcal{Z}} p(d) \geq \Gamma$. When p is heavy-tailed (Appendix H), this constraint is satisfied by a short-range prefix $\mathcal{Z} = \{1, \ldots, k\}$ with $k \ll \Gamma|\mathcal{D}|$. The retained set $\mathcal{Z}$ thus excludes the bulk of distance buckets ($|\mathcal{D} \setminus \mathcal{Z}|$ approaches $|\mathcal{D}|$), so the per-bucket guarantee of Corollary G.8 is lost. At test time, any query whose evidence falls in an excluded long-distance bucket incurs an output error proportional to the mass on that bucket, with no protection from the calibration.*

Interpretation. Proposition G.7 together with Corollary G.8 promotes Theorem G.1 from an algebraic identity to an attention-output approximation guarantee: under IPW, the calibration-time row-averaged output error is bounded by $2C \cdot (1-\Gamma)|\mathcal{D}|\mu_{\max}/(L\,|\mathcal{D}_{\text{calib}}|)$, with the bound spread *evenly* across distance buckets. Corollary G.9 shows this even spread is exactly what unweighted saliency cannot deliver, regardless of the absolute retention level. The bound is on attention output rather than task loss, but it isolates the precise sense in which IPW gives a stronger reconstruction guarantee than $\varphi \equiv 1$ at equal retention level.

Remark G.10 (Bucketization dependence). Both Theorem G.1 and Proposition G.7 are stated for a fixed bucketization $\mathcal{B}$. Different choices of $\mathcal{B}$ yield different IPW reweightings $\varphi^{\star}_{\mathcal{B}}$, but the uniqueness conclusion holds for each $\mathcal{B}$ independently. Empirically, calibration outcomes are insensitive to the precise bucketization (Appendix E, robustness ablation): varying the bucket count over $\{16, 32, 64, 128, 256\}$ produces LongBench scores within ± 0.3 point and average bit-widths within ± 0.05.

H Attention weight distribution across relative distances

To estimate $p^{(l,h)}(d)$ robustly, we perform *distance bucketing* on a calibration set $\mathcal{S}$. Let the bucket width be Δ; relative distances are grouped by

$$k \;=\; \big\lfloor d \,/\, \Delta \big\rfloor, \qquad \mathcal{B}_k \;=\; \{\, d \;\mid\; k\Delta \leq d < (k+1)\Delta \,\}. \tag{XII}$$

For each (l, h), we first aggregate the *attention weight mass* in bucket $\mathcal{B}_k$ across the calibration set:

$$M_k^{(l,h)} \;=\; \sum_{x \in \mathcal{S}} \sum_{i \geq j,\; d_{i,j} \in \mathcal{B}_k} \mathbf{W}_{i,j}^{(l,h)}(x), \tag{XIII}$$

and normalize to obtain the empirical *attention-weight distribution over distance*:

$$p^{(l,h)}(k) \;=\; \frac{M_k^{(l,h)}}{\sum_{k'} M_{k'}^{(l,h)}}. \tag{XIV}$$

We then define $p^{(l,h)}(d) \equiv p^{(l,h)}(k)$ for $d \in \mathcal{B}_k$. Intuitively, this procedure does *not* count pairs; instead, it measures how much attention mass is placed at each distance, bucketed by d. Long-range buckets that are rarely attended (low $p^{(l,h)}$) are upweighted by IPW. The factor L_{ctx} keeps the magnitude of φ_{IPW} on the same scale as RDW.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the core methodological contributions (block-aligned mixed-precision attention, affine windows, IPW-based calibration, and layer-adaptive regularization) and summarize the main empirical claim using the same scope and metrics as the experiments; see the Abstract, Section 5.1, and Section 5.2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer:[Yes]

   Justification: We discuss methodological limitations in the conclusion, including that the current formulation is designed around causal-distance structure and is most natural for causal self-attention; see Section 5.2 for empirical scope and Section 5.1 plus Section 6 for assumptions and limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The assumptions for the IPW characterization, scaling invariance, and output-error bound are stated in the main text and appendix, and full proofs are provided in Appendix G and Appendix G.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer:[Yes]

   Justification: We specify models, benchmarks, baselines, calibration data, calibration sequence length, sink-token count, bucket count, hardware, and key hyperparameters in Section 5.1, with additional implementation, robustness, and appendix tables in Appendix D and Appendix E. We also upload the implementation of our kernel in supplementary materials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We upload the implementation of our kernel in supplementary materials.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the evaluated models, datasets, metrics, baselines, calibration protocol, hyperparameters, and inference hardware in Section 5.1; additional hyperparameter and robustness details are given in Appendix D and Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report deterministic single-run evaluations without error bars or significance tests; this is stated implicitly by the fixed evaluation setup, and repeated long-context benchmark runs were computationally prohibitive for this submission.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the GPU types used for accuracy, kernel, and end-to-end experiments, and we provide calibration memory and time in Section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, the research complies with the NeurIPS Code of Ethics. The work is an algorithmic efficiency study using existing public models and benchmarks and does not involve human subjects or sensitive private data collection.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper focuses on algorithmic efficiency and does not include a dedicated broader-impact discussion. Potential positive impact includes reduced inference cost and energy use; potential negative impact is that more efficient long-context inference could also lower the deployment cost of misuse.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The submission does not release a new pre-trained language model, dataset, or public artifact package that would require misuse safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: We cite the models, datasets, and baseline systems used in the experiments, but we do not explicitly enumerate licenses and terms of use for each external asset in the paper.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The submission does not release a new dataset, model checkpoint, or code artifact alongside the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not describe potential risks incurred by study participants.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer:[N/A]

Justification: LLMs are the evaluation targets of the paper, but they are not used as a non-standard component in the core method development itself.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.