SPARSE ATTENTION IS MATRIX APPROXIMATION, NOT CHOOSING FROM A BAG OF VALUES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) achieve strong performance across many domains, but their efficiency is limited by the quadratic cost of attention with respect to prompt length. Sparse attention reduces this cost by retaining only a small fraction of query-key interactions to approximate the full attention matrix. However, existing methods are trapped in a mathematically wrong view: they simply keep large scalar entries or high-mass regions of the attention matrix. This treats the attention matrix as a bag of values, ignoring that it is used as a structured matrix whose entries jointly determine the attention output through multiplication with value vectors. We argue that this is the core conceptual issue: **sparse attention should be formulated as matrix approximation, not as blindly choosing the largest values from a bag of entries.** Based on this view, we propose **M**atrix **A**pproximation **S**parse **A**ttention (**MASA**). MASA replaces raw attention-mass ranking with a closed-form score that measures how much each sparse unit reduces matrix-product approximation error. As a theory-grounded plug-in correction, MASA can be added to existing sparse attention frameworks without changing their sparse kernels or budgets. Extensive experiments across multiple sparse attention methods, benchmarks, and LLM backbones show consistent accuracy gains, supporting both MASA and the matrix-approximation view of sparse attention.

1 INTRODUCTION

Large Language Models (LLMs) have achieved strong performance across a wide range of applications, from natural language processing (Radford et al., 2018) to mathematical reasoning (Ahn et al., 2024), multimodal tasks (Team et al., 2023), and scientific discovery (Zhang et al., 2025b). However, efficiency remains a central challenge for LLMs (Bai et al., 2024a). As tasks become more complex and prompts become more structured, the input length continues to grow rapidly (Liu et al., 2025). Since attention computation has quadratic complexity with respect to the prompt length (Vaswani et al., 2017), attention becomes a major bottleneck for efficient long-context inference.

Sparse attention is a natural way to reduce this cost while keeping the main modeling benefit of attention. Prior work has observed that attention matrices often contain sparse and structured patterns (Jiang et al., 2024a), suggesting that only a small fraction of query-key interactions is needed to approximate full attention. Based on this observation, many methods select a sparse subset of the attention matrix under a compute budget. Early approaches use fixed or predefined patterns: Sliding Window Attention (Jiang et al., 2023a) restricts each query to a local context window, while MInference (Jiang et al., 2024a) approximates attention using vertical lines, slash lines, and blocks. Other methods make the selection input-aware: SeerAttention (Gao et al., 2025) learns to predict block-sparse attention patterns, and Quest (Tang et al., 2024) retrieves query-aware key pages. More recent methods further adapt the sparse budget, such as FlexPrefill (Lai et al., 2025), which decides when to stop selecting additional sparse units, and Tactic (Zhu et al., 2025), which adjusts the budget through clustering and distribution fitting. **Despite these different designs, the final selection rule remains ultimately magnitude-driven: sparse attention methods keep large scalar entries or high-mass regions of the attention matrix.**

This selection principle has a mathematical mismatch. As illustrated in Fig. 1, a candidate with much larger mass can preserve the matrix product substantially worse under the same sparse budget. Magnitude-based selection treats the attention matrix as a bag of scalar values, whereas Transformers use it as a structured matrix acting on value vectors through matrix multiplication (Vaswani et al., 2017). Therefore, the right question is not which entries of the attention matrix are largest, but which selected entries best preserve the matrix product that produces the attention output. In other words,

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

sparse attention should be formulated as matrix approximation, not as blindly choosing the largest values from a bag of entries.

Based on this view, we propose **M**atrix **A**pproximation **S**parse **A**ttention (**MASA**), a theory-grounded plug-in correction for sparse attention. MASA keeps the sparse units, kernels, and budgets of existing sparse attention methods, but replaces their raw attention-mass ranking with a closed-form score derived from matrix-product approximation error. This changes the selection target from preserving large entries of the attention matrix to preserving the matrix action that determines the attention output. As a result, MASA can be added to diverse sparse attention frameworks without changing their implementation interface.

In this work, our contributions are summarized as follows:

- **Theory.** We identify a long-standing selection mismatch in sparse attention: existing magnitude-based rules approximate the attention matrix entrywise, while the Transformer uses attention through matrix multiplication. We provide a matrix-approximation view that gives the correct selection objective.
- **Method.** We propose MASA, a theory-grounded plug-in scoring method that ranks sparse units by their closed-form reduction in matrix-product approximation error. MASA can be integrated into existing sparse attention baselines while keeping their sparse kernels and budgets unchanged.
- **Experiments.** We evaluate MASA across multiple sparse attention methods, benchmarks, and LLM backbones. Under matched settings with the original baselines, MASA gives consistent accuracy gains, supporting both the proposed method and the matrix-approximation view of sparse attention.

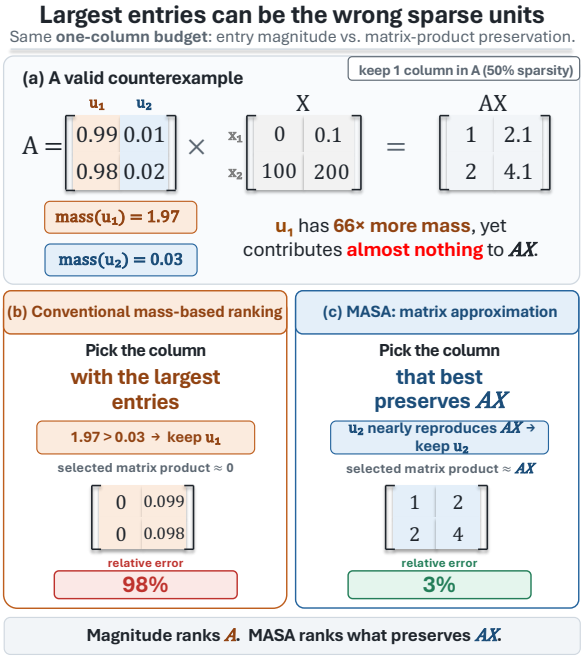


Figure 1: A minimal two-column counterexample. Under the same one-column budget, conventional mass-based ranking selects u_1, whose mass is $66\times$ larger but incurs 98% relative matrix-product error. MASA selects u_2 and reduces the error to 3%. Here, $\boldsymbol{X}$ is an illustrative probe matrix.

2 PRELIMINARIES

We review sparse attention methods through three representative frameworks used to evaluate MASA, then formalize sparse attention. Broader LLM acceleration methods are discussed in Appendix A.

2.1 RELATED WORK

MInference (Jiang et al., 2024a) is a training-free framework that exploits **multiple structured patterns through offline head-wise assignment**. It searches among A-shaped, vertical-slash, and block-sparse patterns for each attention head. The assigned pattern type and budget remain fixed, while sparse indices are constructed dynamically for each input. MoA (Fu et al., 2025) similarly uses offline configuration, but sets head- and layer-specific sliding-window spans.

SeerAttention (Gao et al., 2025) instead **learns input-dependent sparse layouts within a block-based representation**. GPU-friendly blocks can approximate different attention patterns without explicit head-wise pattern-type assignment. A lightweight gate learns block importance from dense attention supervision, providing a representative learned selector under the fixed budget used in our evaluation. SpargeAttn (Zhang et al., 2025a) provides a training-free alternative using online attention prediction and softmax-aware filtering.

FlexPrefill (Lai et al., 2025) further increases flexibility by **adapting both pattern type and budget without additional training**. For each input and head, it chooses between vertical-slash and query-aware block selection, then determines the number of retained units using a cumulative-attention

threshold. XAttention (Xu et al., 2025) also performs input-dependent block selection, estimating block importance through antidiagonal sampling.

Together, MInference, SeerAttention, and FlexPrefill cover the main design choices in sparse attention: vertical, slash, and block units; offline-assigned and input-adaptive pattern types; fixed and adaptive budgets; and training-free and learned selection. Their complementary mechanisms provide a suitable testbed for evaluating *whether MASA gives a general scoring correction across distinct sparse attention designs, rather than an improvement confined to a single implementation.*

2.2 Formalization

In attention, $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are the query, key, and value matrices, where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{N\times d}$. Their height is the prompt sequence length N, which can grow very large and become the main bottleneck in attention computation. After applying the causal mask, the full causal attention matrix and output are

$$\boldsymbol{A} = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}} + \boldsymbol{M}\right), \qquad \boldsymbol{O} = \boldsymbol{A}\boldsymbol{V}, \qquad \boldsymbol{M}_{ij} = \begin{cases} 0, & j \le i, \\ -\infty, & j > i. \end{cases}$$

Here $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is the full causal attention matrix.

However, full causal attention still computes attention scores for all valid query-key pairs, leading to $\mathcal{O}(N^2)$ complexity. Sparse attention reduces this cost by retaining only a selected subset of valid query-key pairs. Let $\mathcal{S}$ denote the sparse attention index set, and let $\boldsymbol{M}_{\mathcal{S}} \in \{0,1\}^{N\times N}$ be its binary retention mask defined by $\boldsymbol{M}_{\mathcal{S}}[i,j] = \mathbf{1}[(i,j) \in \mathcal{S}]$.

Since $\mathcal{S}$ only contains valid causal positions, the causal constraint is already included in the definition of $\mathcal{S}$. For selection analysis, we write the sparse attention contribution as

$$\boldsymbol{O}_{\mathcal{S}} = (\boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{V}.$$

This notation isolates the retained support $\mathcal{S}$. In actual sparse-attention implementations, the selected support is passed to the baseline kernel, which may apply its own sparse-softmax normalization or other kernel-specific details. MASA does not modify this sparse computation; it only changes how $\mathcal{S}$ is ranked before the baseline kernel is called.

Normal causal attention is the special case where $\mathcal{S} = \{(i,j) \mid 1 \le j \le i \le N\}$.

3 Method

We first identify the selection mismatch shared by existing sparse attention methods in Sec. 3.1. We then introduce MASA as a matrix-approximation correction in Sec. 3.2, and discuss its implementation in the sparse prefilling setting in Sec. 3.3.

3.1 Issue

The goal of sparse attention is to reduce the number of selected query-key pairs while keeping the sparse attention output close to the full attention output. This can be written as a two-objective problem (Lai et al., 2025):

$$\min_{\mathcal{S}} \|\boldsymbol{A}\boldsymbol{V} - (\boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{V}\|_F^2, \qquad \min_{\mathcal{S}} |\mathcal{S}|. \tag{1}$$

The first objective measures the output error between full and sparse attention, and the second objective measures the sparse attention cost. Here $\|\cdot\|_F^2$ is the squared Frobenius norm, which sums squared errors over all entries of a matrix; see Appendix B. We use this form because it gives a closed-form MASA score, while more general matrix-action metrics are discussed in Appendix C.

Under a fixed compute budget B, the sparse selection problem in Eq. (1) becomes

$$\mathcal{S}^\star = \arg\min_{\mathcal{S}} \|\boldsymbol{A}\boldsymbol{V} - (\boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{V}\|_F^2 \quad \text{s.t.} \quad |\mathcal{S}| \le B. \tag{2}$$

This is the objective implied by attention computation: the selected sparse pattern should preserve the matrix product $\boldsymbol{A}\boldsymbol{V}$.

Current sparse selection. Most existing sparse attention methods do not directly estimate the output error in Eq. (2). In sparse prefill implementations, they first define candidate sparse units, such

as blocks, vertical lines, slash lines, or query-aware index groups, as discussed in Sec. 2.1. For a candidate unit u, they assign a scalar entrywise score

$$s_{\text{entry}}(u) = \sum_{(i,j)\in u} \phi(\boldsymbol{A}_{ij}), \tag{3}$$

where ϕ maps each attention entry to a scalar score. In most cases, ϕ corresponds to attention mass, but it can also be maximum mass, thresholded mass, or a learned gate-score variant (Gao et al., 2025). The sparse pattern is then formed by keeping the units with the largest $s_{\text{entry}}(u)$ under the budget. This rule reduces the attention matrix to entry scores $\boldsymbol{A} \longmapsto \{\phi(\boldsymbol{A}_{ij})\}_{i,j}$. The corresponding entrywise approximation loss is

$$\|\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A}\|_F^2 = \sum_{(i,j)\notin\mathcal{S}} \boldsymbol{A}_{ij}^2.$$

Since $\sum_{i,j} \boldsymbol{A}_{ij}^2$ is fixed, minimizing this loss under a fixed budget is equivalent to keeping the largest squared attention entries:

$$\arg\min_{|\mathcal{S}|\le B} \|\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A}\|_F^2 = \arg\max_{|\mathcal{S}|\le B} \sum_{(i,j)\in\mathcal{S}} \boldsymbol{A}_{ij}^2. \tag{4}$$

Therefore, current sparse selection asks which entries or regions of $\boldsymbol{A}$ are large. It does not ask which selected entries best preserve the actual attention output $\boldsymbol{AV}$.

Mismatch. The Transformer does not consume the attention matrix as isolated scalar entries. It uses the attention matrix as a matrix acting on value vectors. From Eq. (2), the true output error is

$$\|\boldsymbol{AV} - (\boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{V}\|_F^2 = \left\|\left(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A}\right)\boldsymbol{V}\right\|_F^2.$$

Expanding this matrix-product error gives

$$\left\|\left(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A}\right)\boldsymbol{V}\right\|_F^2 = \operatorname{tr}\left(\left(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A}\right)\boldsymbol{V}\boldsymbol{V}^\top\left(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A}\right)^\top\right). \tag{5}$$

This shows that **entrywise magnitude ranking is correct only when $\boldsymbol{V}\boldsymbol{V}^\top$ behaves like an identity matrix** over token positions. This condition does not hold in long-context attention. Since $\boldsymbol{V} \in \mathbb{R}^{N\times d}$ and usually $N \gg d$,

$$\operatorname{rank}(\boldsymbol{V}\boldsymbol{V}^\top) \le d \ll N.$$

Therefore, $\boldsymbol{V}\boldsymbol{V}^\top$ cannot be an identity matrix over the N token positions.

This is the core mismatch. Existing sparse attention methods mainly answer: *which attention entries are large?* The matrix-approximation problem asks a different question: *which selected entries best preserve $\boldsymbol{AV}$?* These two questions are not equivalent.

3.2 MASA

Main idea. The objective in Eq. (2) shows that sparse attention should preserve the matrix product $\boldsymbol{AV}$, rather than only large entries of $\boldsymbol{A}$. We propose **M**atrix **A**pproximation **S**parse **A**ttention (**MASA**), a plug-in scoring correction for sparse attention. MASA keeps the same candidate sparse units and the same budget as the baseline, such as blocks, vertical lines, slash lines, or query-aware index groups. It only replaces the score used to rank these units.

Let $\boldsymbol{X} \in \mathbb{R}^{N\times r}$ be a probe matrix. The dense probe output and sparse probe output are

$$\boldsymbol{O}_{\boldsymbol{X}} = \boldsymbol{AX}, \qquad \boldsymbol{O}_{\boldsymbol{X},\mathcal{S}} = (\boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{X}.$$

MASA scores sparse patterns by the matrix-product approximation error

$$\mathcal{L}_{\text{MASA}}(\mathcal{S}; \boldsymbol{X}) = \|\boldsymbol{O}_{\boldsymbol{X}} - \boldsymbol{O}_{\boldsymbol{X},\mathcal{S}}\|_F^2 = \|(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{X}\|_F^2. \tag{6}$$

When $\boldsymbol{X} = \boldsymbol{V}$, Eq. (6) becomes exactly the attention output error in Eq. (2). Thus, MASA changes sparse selection from preserving large entries of $\boldsymbol{A}$ to preserving the matrix product $\boldsymbol{AX}$. Fig. 1 gives a minimal two-column instance of this objective: although u_2 has much lower mass, it receives a higher MASA score because it preserves $\boldsymbol{AX}$ substantially more accurately. **We use the general notation $\boldsymbol{X}$ rather than $\boldsymbol{V}$ because the probe can also be pooled, norm-controlled, or projected value information, as discussed in Sec. 3.3.**

Expanding Eq. (6) gives

$$\mathcal{L}_{\text{MASA}}(\mathcal{S}; \boldsymbol{X}) = \operatorname{tr}\left((\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{X}\boldsymbol{X}^\top(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})^\top\right).$$

This has the same form as Eq. (5), with $\boldsymbol{X}\boldsymbol{X}^\top$ replacing $\boldsymbol{V}\boldsymbol{V}^\top$. Therefore, entrywise magnitude ranking appears only as the identity-Gram special case.

Reduction score. Directly optimizing Eq. (6) over all sparse patterns is combinatorial, so sparse attention methods usually rank candidate units one by one. MASA keeps this ranking interface. For a candidate sparse unit u, let $\boldsymbol{M}_u \in \{0,1\}^{N\times N}$ be its binary retention mask. Its contribution to the probe output is $\boldsymbol{C}_u = (\boldsymbol{M}_u \odot \boldsymbol{A})\boldsymbol{X}$. MASA scores u by the reduction in squared matrix-product error obtained by retaining this contribution:

$$s_{\mathrm{MASA}}(u) = \|\boldsymbol{O}_{\boldsymbol{X}}\|_F^2 - \|\boldsymbol{O}_{\boldsymbol{X}} - \boldsymbol{C}_u\|_F^2 = 2\langle \boldsymbol{O}_{\boldsymbol{X}}, \boldsymbol{C}_u\rangle_F - \|\boldsymbol{C}_u\|_F^2. \tag{7}$$

The sparse pattern is formed by keeping the units with the largest $s_{\mathrm{MASA}}(u)$ under the same budget as the baseline. If the budget is adaptive, units with non-positive scores can simply be skipped. Compared with the entrywise score in Eq. (3), **MASA asks whether a unit helps reconstruct $\boldsymbol{A}\boldsymbol{X}$, not whether the entries inside the unit are large.**

For a single query-key pair (i, j), let $\boldsymbol{O}_{\boldsymbol{X},i}$ and $\boldsymbol{X}_j$ denote the corresponding row vectors of $\boldsymbol{O}_{\boldsymbol{X}}$ and $\boldsymbol{X}$. Then

$$s_{\mathrm{MASA}}(i,j) = 2\boldsymbol{A}_{ij}\,\langle \boldsymbol{O}_{\boldsymbol{X},i}, \boldsymbol{X}_j\rangle - \boldsymbol{A}_{ij}^2\,\|\boldsymbol{X}_j\|_2^2\,. \tag{8}$$

This closed form shows why attention magnitude alone is insufficient. The score also depends on the direction and norm of the probe vector $\boldsymbol{X}_j$, and on its alignment with the dense matrix action $\boldsymbol{O}_{\boldsymbol{X},i}$.

The connection to entrywise ranking follows directly from Eq. (8). If $\boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{I}$, then

$$\langle \boldsymbol{O}_{\boldsymbol{X},i}, \boldsymbol{X}_j\rangle = \boldsymbol{A}_{ij}, \qquad \|\boldsymbol{X}_j\|_2^2 = 1, \qquad s_{\mathrm{MASA}}(i,j) = \boldsymbol{A}_{ij}^2.$$

Thus, squared-magnitude ranking is only the identity-Gram special case of MASA. As discussed in Sec. 3.1, this condition does not hold for value-derived probes in long-context attention.

3.3 Implementation

Our theory applies to sparse attention selection in general. In our implementation and experiments, we instantiate MASA in the sparse prefill setting, where the main cost comes from computing attention over a long prompt before decoding. This is also the setting of our baselines, including MInference (Jiang et al., 2024a), SeerAttention (Gao et al., 2025), and FlexPrefill (Lai et al., 2025). MASA keeps their sparse unit definitions, sparse kernels, and budget rules unchanged, and only replaces the score used to rank candidate sparse units.

All MASA scores are computed at the same profile resolution as the baselines. Thus, $\boldsymbol{A}$ and $\boldsymbol{X}$ in the scoring formulas denote the baseline's profile attention map and the matching profile probe when the baseline ranks units at block or other reduced resolution. MASA therefore does not require a new full-resolution dense attention map beyond the information used for sparse-unit ranking.

Structured sparse units. MASA can be used with the same sparse unit family as the baseline. For any candidate unit u, the exact score is the reduction score in Eq. (7). When a unit contains at most one key position per query row, this score reduces to the sum of pair scores:

$$s_{\mathrm{MASA}}(u) = \sum_{(i,j)\in u} s_{\mathrm{MASA}}(i,j).$$

This applies to vertical and slash units (Jiang et al., 2024a; Lai et al., 2025):

$$s_{\mathrm{MASA}}(j) = \sum_i s_{\mathrm{MASA}}(i,j), \qquad s_{\mathrm{MASA}}(\ell) = \sum_{i-j=\ell} s_{\mathrm{MASA}}(i,j).$$

For block units, the full score in Eq. (7) should be used because multiple key positions in the same query row interact through the squared norm term. If I and J are the query and key index ranges of a block, the block score is

$$s_{\mathrm{MASA}}(I,J) = \sum_{i\in I}\left[2\left\langle \boldsymbol{O}_{\boldsymbol{X},i}, \sum_{j\in J}\boldsymbol{A}_{ij}\boldsymbol{X}_j\right\rangle - \left\|\sum_{j\in J}\boldsymbol{A}_{ij}\boldsymbol{X}_j\right\|_2^2\right]. \tag{9}$$

The second term keeps the interaction among entries inside the same block, so the block is scored by matrix-product error reduction rather than by entry magnitude.

Memory-safe computation. A direct implementation of Eq. (8) may seem to require all vectors $\boldsymbol{A}_{ij}\boldsymbol{X}_j$, which would form an $N \times N \times r$ tensor. MASA avoids this tensor. First compute

$$\boldsymbol{O}_{\boldsymbol{X}} = \boldsymbol{A}\boldsymbol{X}, \qquad \boldsymbol{C} = \boldsymbol{O}_{\boldsymbol{X}}\boldsymbol{X}^{\top}, \qquad \mathbf{n}_j = \|\boldsymbol{X}_j\|_2^2.$$

Then all pair scores are obtained by

$$\boldsymbol{D} = 2\boldsymbol{A} \odot \boldsymbol{C} - \boldsymbol{A}^{\odot 2} \odot (\mathbf{1}\mathbf{n}^{\top}),$$

where $\boldsymbol{D}_{ij} = s_{\mathrm{MASA}}(i, j)$, $\boldsymbol{A}^{\odot 2}$ denotes entrywise square, and $\mathbf{1}$ is the all-one column vector. This computation only uses matrix products and score matrices at the profile resolution, without materializing the $N \times N \times r$ contribution tensor.

For vertical and slash units, we aggregate entries of $\boldsymbol{D}$ according to the unit definition of the baseline. For block units, we use Eq. (9) and compute the block contribution $\sum_{j \in J} \boldsymbol{A}_{ij}\boldsymbol{X}_j$ within the block. This keeps the scoring memory-safe while retaining the block-level reduction score.

Probe choice. The direct probe is $\boldsymbol{X} = \boldsymbol{V}$, because the true attention output is $\boldsymbol{A}\boldsymbol{V}$. Raw value vectors, however, can contain high-norm outliers. Since attention output depends on both attention weights and value vectors (Kobayashi et al., 2020), large value norms can dominate the score in Eq. (8). Similar outlier effects have also been shown to harm Transformer computation and quantization robustness (Bondarenko et al., 2023). We therefore use norm-controlled value probes to keep value directions while reducing the effect of extreme norms.

One option is soft norm balancing:

$$\boldsymbol{X}_j = \frac{\boldsymbol{V}_j}{(\|\boldsymbol{V}_j\|_2 + \epsilon)^{\alpha}}, \qquad 0 \le \alpha \le 1. \tag{10}$$

When $\alpha = 0$, this is the raw value probe. When $\alpha = 1$, it keeps the value direction while removing the value norm. Intermediate values keep part of the norm information while reducing the effect of high-norm outliers.

Another option is norm clipping:

$$\boldsymbol{X}_j = \boldsymbol{V}_j \cdot \min\left(1, \frac{c}{\|\boldsymbol{V}_j\|_2 + \epsilon}\right), \tag{11}$$

where c is a chosen percentile of value-vector norms in the profile region. This keeps normal value vectors unchanged and only clips unusually large norms.

The probe view also **supports more general matrix-action metrics**. Appendix C shows that weighted output metrics can be written in the same form by choosing an appropriate probe $\boldsymbol{X}$, and that the Frobenius matrix-product error controls the spectral matrix-product error. Finally, when the baseline ranks units on a profile attention map, we build $\boldsymbol{X}$ at the same profile resolution. For example, block-level selection uses block-pooled value probes, so that the probe **matches the sparse units used by the implemented kernel**.

4 Experiments

Existing sparse attention studies often differ in budgets, kernels, LLM backbones, and benchmark choices, which makes isolated comparisons hard to interpret. We therefore evaluate MASA under a controlled protocol designed to test whether the proposed matrix-approximation view improves existing sparse attention methods **without changing their own settings**. First, we do not evaluate MASA on a single favorable framework, but attach it to three representative sparse attention baselines with different sparse units and selection rules. Second, we do not tune the sparse budget, kernel, baseline hyperparameters, or data processing for MASA; each baseline keeps its official configuration, and MASA only replaces the sparse-unit ranking score. Third, we evaluate on the long-context benchmarks used by the original baselines, including RULER (Hsieh et al., 2024), InfiniteBench (Zhang et al., 2024), and LongBench (Bai et al., 2024b). This protocol directly tests **whether MASA improves sparse attention under the conditions where the baselines are already designed to work well**.

4.1 Setups

Baselines. We evaluate MASA by augmenting three representative sparse attention methods: MInference (Jiang et al., 2024a), SeerAttention (Gao et al., 2025), and FlexPrefill (Lai et al., 2025), as

Table 1: RULER (Hsieh et al., 2024) results under matched sparse-attention settings. Within each framework, Original and MASA use the same sparse implementation and differ only in the ranking rule; Full Attention is the reference. Δ_{Full} is the signed Avg. difference from Full Attention. Higher is better. **Bold** marks the better Avg. between Original and MASA.

		Variant	4K	8K	16K	32K	64K	128K	Avg.	Δ_{Full}
MInfer		Full Attention	97.2	91.8	87.3	80.8	77.4	72.2	84.4	0
		Original	97.7	91.2	88.5	85.0	82.3	77.6	87.0	+2.6
		MASA	97.93	91.29	88.50	85.45	81.57	78.88	**87.27**	**+2.87**
SeerAttn		Full Attention	95.53	92.37	92.01	87.63	84.39	76.26	88.01	0
		Original	95.53	92.71	92.02	88.49	83.48	73.37	87.60	-0.41
		MASA	95.53	92.74	92.84	89.56	84.01	73.85	**88.09**	**+0.08**
FlexPrefill	*Llama*	Full Attention	95.67	93.75	93.03	87.26	84.37	78.13	88.70	0
		Original	95.43	93.51	94.71	89.42	82.93	79.09	89.18	+0.48
		MASA	95.43	93.99	94.95	89.66	85.09	80.54	**89.94**	**+1.24**
	Qwen	Full Attention	89.90	88.70	80.77	79.33	56.49	17.79	68.83	0
		Original	90.39	89.91	83.17	81.25	59.14	20.67	70.75	+1.92
		MASA	90.88	88.94	84.85	82.64	62.79	22.83	**72.16**	**+3.33**

Table 2: InfiniteBench (Zhang et al., 2024) results under matched sparse-attention settings. Within each framework, Original and MASA use the same sparse implementation and differ only in the ranking rule; Full Attention is the reference. Δ_{Full} is the signed Avg. difference from Full Attention. Higher is better. **Bold** marks the better Avg. between Original and MASA.

		Variant	En.Sum	En.QA	En.MC	En.Dia	Zh.QA	Code Debug	Math Find	Retr. PassKey	Retr. Number	Retr. KV	Avg.	Δ_{Full}
MInfer		Full Attention	20.2	12.4	67.3	6.0	12.9	22.1	26.6	100.0	100.0	14.4	38.2	0
		Original	20.5	12.9	65.9	7.5	12.5	22.3	33.1	100.0	100.0	12.8	38.8	+0.6
		MASA	19.73	12.15	66.81	8.50	12.28	22.84	33.43	100.00	100.00	16.20	**39.20**	**+1.00**
FlexPrefill	*Llama*	Full Attention	31.91	25.92	69.43	21.50	31.95	16.75	24.29	99.15	99.66	60.00	48.06	0
		Original	31.82	24.82	69.43	19.50	35.46	16.75	31.14	98.64	99.83	44.00	47.14	-0.92
		MASA	32.10	26.25	66.81	19.50	33.66	20.05	29.14	98.31	99.66	54.40	**47.99**	**-0.07**
	Qwen	Full Attention	4.65	5.57	34.50	9.00	11.27	24.87	24.57	95.42	75.42	0.00	29.53	0
		Original	14.27	6.55	34.06	13.50	11.51	20.81	30.86	96.95	72.20	0.00	30.07	+0.54
		MASA	15.55	6.38	32.31	16.50	13.09	21.07	30.29	94.41	74.58	0.00	**30.42**	**+0.89**

discussed in Sec. 2.1. These baselines cover vertical, slash, and block-wise sparse patterns; fixed and input-adaptive selection; fixed and adaptive budgets; and training-free and learned selection rules. This makes them a strong test bed for evaluating whether MASA is a general correction rather than a method-specific modification.

Implementation. All experiments are conducted on a single NVIDIA H100 80GB GPU with CUDA 12.6. The FlashAttention (Dao et al., 2022) and Triton (Tillet et al., 2019) settings follow the corresponding baseline implementations. For each sparse attention baseline, MASA uses the official code path, released weights, sparse kernels, sparse unit definitions, and budget rules. The only change is the ranking score used to select candidate sparse units. This matched setting isolates the effect of the MASA score from other engineering factors.

Benchmarks. We follow the benchmark choices of the original baselines. RULER (Hsieh et al., 2024) is used for all three baselines. It is a synthetic long-context benchmark with controllable sequence lengths and task types, including needle-in-a-haystack retrieval, multi-hop tracing, and aggregation. InfiniteBench (Zhang et al., 2024) is used for MInference and FlexPrefill. It evaluates long-dependency understanding over contexts with an average of about 214K tokens and includes both synthetic and real-world tasks. LongBench (Bai et al., 2024b) is used for SeerAttention. It is a bilingual long-context benchmark covering multi-document question answering, summarization, and retrieval tasks. Using the baselines' own benchmark settings avoids choosing tasks that favor MASA.

LLM Backbones. Following the baseline settings, we use Llama-3-8B-Instruct-262k for MInference, Llama-3.1-8B-Instruct for SeerAttention, and both Llama-3.1-8B-Instruct and Qwen2-7B-Instruct for FlexPrefill. These test MASA across different LLM backbones rather than a single model family.

4.2 Main Results

Tables 1–3 summarize the main results; Appendix D provides the full comparisons. Within each framework, Original and MASA use the same LLM backbone, benchmark, candidate units, sparse kernel, and sparse budget, differing only in the ranking score. Full Attention provides the dense reference, and Δ_{Full} reports the signed average-score difference from it. Each Avg. summarizes the corresponding task or length set. **MASA improves all eight matched average scores across three sparse methods, three benchmark suites, and both Llama- and Qwen-family backbones.**

These gains are obtained under the baselines' released configurations, where the original sparse methods already approach or exceed the dense reference.[1] **MASA nevertheless delivers substantial gains relative to the dense-reference differences in several settings.** On InfiniteBench, MASA raises FlexPrefill-Llama from 47.14 to 47.99 against a full-attention score of 48.06, recovering approximately 92% of the original average-score deficit. On RULER, MASA raises SeerAttention from 87.60 to 88.09, moving it from below the dense reference to above it. For FlexPrefill-Llama on RULER, replacing the ranking score adds 0.76 points, exceeding the original method's 0.48-point advantage over full attention and increasing that advantage to 1.24 points. Thus, MASA improves **already strong sparse systems** without increasing their sparse budgets or changing their kernels.

The benefits are particularly evident at longer context lengths, the main target of sparse attention acceleration. Across the four RULER configurations at 32K, 64K, and 128K, MASA improves 11 of 12 matched length-level scores. For every configuration, the mean gain over these longer contexts at 32K–128K exceeds that at 4K–16K. At 128K, MASA improves MInference, SeerAttention, FlexPrefill-Llama, and FlexPrefill-Qwen by 1.28, 0.48, 1.45, and 2.16 points, respectively. At 64K, the gains reach 2.16 points for FlexPrefill-Llama and 3.65 points for FlexPrefill-Qwen. On LongBench, MASA improves SeerAttention's 8K+ score from 52.69 to 53.16. These results support the value of matrix-approximation ranking for long-context sparse prefill under the original frameworks' sparse budgets and kernels.

Table 3: LongBench (Bai et al., 2024b) results under matched sparse-attention settings. Within SeerAttention, Original and MASA differ only in the sparse-unit ranking rule; Full Attention is the reference. Δ_{Full} is the signed Avg. difference from Full Attention. Higher is better. **Bold** marks the better Avg. and its Δ_{Full}.

Variant	0–4K	4–8K	8K+	Avg.	Δ_{Full}
Full Attention	55.32	53.98	52.90	54.07	0
Original	55.43	54.49	52.69	54.20	+0.13
MASA	55.45	54.48	53.16	**54.37**	**+0.30**

Beyond downstream accuracy, Appendix E compares executed sparse-attention outputs with dense attention. The evaluated MASA configurations reduce normalized squared output error in all four framework–backbone pairs, providing direct empirical support for matrix-product preservation.

4.3 Efficiency

MASA replaces the baseline's sparse-unit ranking score rather than adding a new attention component. Its only additional computation is evaluating the reduction score in Eq. (8) within the existing ranking stage. We therefore measure kernel-level latency under the same sparse settings to verify that this replacement preserves the efficiency advantage of sparse attention.

Fig. 2 reports latency normalized to dense FlashAttention-2 at sequence lengths 32K, 64K, and 128K. Across MInference, SeerAttention, FlexPrefill-Llama, and FlexPrefill-Qwen, replacing the original ranking score with MASA adds empirically negligible overhead relative to the sparse attention runtime and remains far below the dense attention cost. The overhead also stays negligible at 128K, the main regime where sparse attention is needed. Therefore, MASA improves accuracy without weakening the practical efficiency of the underlying sparse attention method.

4.4 Ablation Study

MASA is derived from the matrix-product approximation objective in Sec. 3.2. The remaining implementation choice is the probe matrix $\boldsymbol{X}$. The direct choice $\boldsymbol{X} = \boldsymbol{V}$ gives the attention-output objective, while Sec. 3.3 also introduces norm-controlled probes to reduce the effect of high-norm value outliers. We ablate soft norm balancing in Eq. (10) and norm clipping in Eq. (11), with full length- and task-level results in Appendix F.

[1] When a sparse baseline already approaches full-attention quality, the remaining score deficit is limited. The relevant question is how much a ranking correction recovers or adds at the same sparse budget, rather than whether it produces a large margin over the dense model. Full attention is a reference, not a strict upper bound on downstream accuracy.

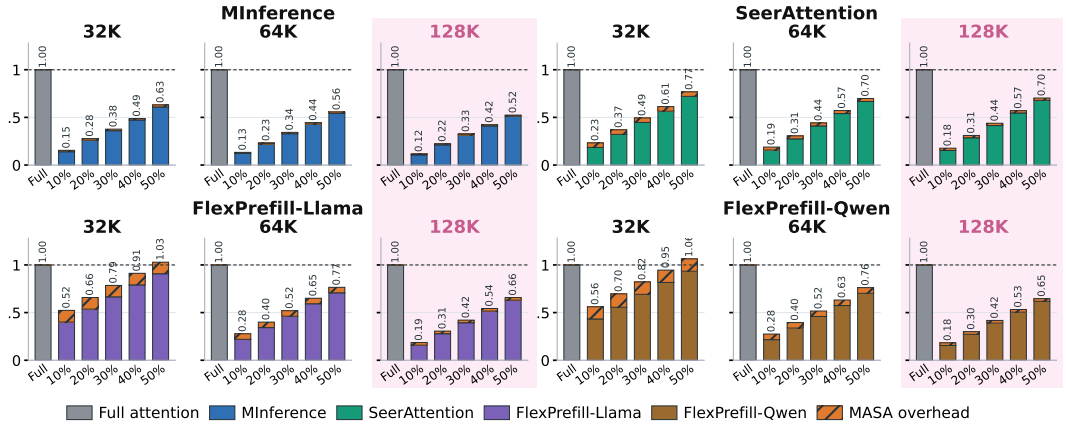


Figure 2: Long-context kernel-level latency normalized to dense FlashAttention-2 at sequence lengths 32K, 64K, and 128K. The x-axis gives the retained sparse budget ratio. All methods use the same latency protocol; method-specific sparse kernels and selection or gating modules follow their official implementations.

Together with the main results in Sec. 4.2, the probe ablation in Table 4 supports two conclusions. First, raw MASA already improves the original sparse baseline in several settings, which supports the matrix-product scoring objective itself. Second, norm control can further improve the score in settings where value-norm outliers affect sparse-unit ranking. Norm clipping is helpful for SeerAttention and FlexPrefill-Llama, while soft norm balancing is useful in some FlexPrefill settings. These trends match the probe design: MASA should preserve value directions while preventing a small number of high-norm value vectors from dominating the selection score.

5 Conclusion

This paper identifies a fundamental selection mismatch in sparse attention. Existing methods usually retain large attention entries or high-mass regions, which treats the attention matrix as a bag of scalar values. However, Transformers use the attention matrix through matrix multiplication with value vectors. Therefore, sparse attention should be formulated as matrix approximation, not as entrywise attention compression. Based on this view, we propose MASA, a theory-grounded plug-in correction that ranks sparse units by their closed-form reduction in matrix-product approximation error. MASA does not change the sparse kernels, candidate sparse units, or budget rules of existing methods, making it orthogonal to low-level attention kernels and other LLM acceleration techniques as discussed in Appendix A. Across multiple sparse baselines, benchmarks, and LLM backbones, MASA gives consistent gains under matched settings with negligible latency overhead. These results support both the proposed method and the broader matrix-approximation view of sparse attention.

Table 4: Ablation of MASA probe choices. Clip p denotes norm clipping at percentile p, and Norm α denotes soft norm balancing. The case $p = -$ and $\alpha = -$ is raw MASA without norm control. Bold marks the best MASA variant for each setting. See details in Sec. 3.3.

	Setting	Clip p	Norm α	Avg.
RULER	MInference	–	–	87.13
	MInference	90	0.6	**87.27**
	SeerAttention	–	–	87.88
	SeerAttention	90	–	**88.09**
	SeerAttention	95	–	87.79
	FlexPrefill-Llama	90	–	89.26
	FlexPrefill-Llama	95	–	89.22
	FlexPrefill-Llama	97.5	–	**89.94**
	FlexPrefill-Qwen	–	–	**72.16**
	FlexPrefill-Qwen	95	–	70.99
InfiniteBench	MInference	–	–	39.05
	MInference	80	0.7	**39.20**
	FlexPrefill-Llama	–	0.5	**47.99**
	FlexPrefill-Qwen	–	–	**30.42**
	FlexPrefill-Qwen	–	0.5	30.31

One practical boundary of MASA is that it corrects the selection score within a given sparse attention framework. It does not search for new sparse kernels, new sparse unit families, or new budget schedules. This is also what makes the comparison clean: MASA isolates the effect of the selection objective while keeping the original sparse attention system unchanged. Future work can combine MASA with better kernel designs, adaptive budget policies, and other LLM acceleration methods, but the core message remains independent of these engineering choices: sparse attention should preserve the matrix action of attention, not blindly keep large scalar entries.

AI USE STATEMENT

LLMs are used to assist with grammar checking, improving writing fluency, managing computational resources and providing suggestions after the initial draft. All observations, ideas, methodologies in this paper are developed entirely by the authors. Any content generated with the help of LLMs is created under detailed author instructions and thoroughly verified by the authors before inclusion.

REFERENCES

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 225–237, 2024.

Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, et al. Beyond efficiency: A systematic survey of resource-efficient large language models. *arXiv preprint arXiv:2401.00625*, 2024a.

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. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024b.

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

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36:75067–75096, 2023.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2978–2988, 2019.

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.

Tianyu Fu, Haofeng Huang, Xuefei Ning, Genghan Zhang, Boju Chen, Tianqi Wu, Hongyi Wang, Zixiao Huang, Shiyao Li, Shengen Yan, et al. Mixture of attention spans: Optimizing llm inference efficiency with heterogeneous sliding-window lengths. In *Second Conference on Language Modeling*, 2025.

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. *Advances in Neural Information Processing Systems*, 2025.

Raghavv Goel, Junyoung Park, Mukul Gagrani, Dalton Jones, Matthew Morse, Harper Langston, Mingu Lee, and Chris Lott. Caote: Kv cache selection for llms via attention output error-based token eviction. *arXiv preprint arXiv:2504.14051*, 2025.

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

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21158–21166, 2024.

Ankit Gupta and Jonathan Berant. Value-aware approximate attention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9567–9574, 2021.

Ke Hong, Guohao Dai, Jiaming Xu, Qiuli Mao, Xiuhong Li, Jun Liu, Kangdi Chen, Yuhan Dong, and Yu Wang. Flashdecoding++: Faster large language model inference with asynchronization, flat gemm optimization, and heuristics. *Proceedings of Machine Learning and Systems*, 6:148–161, 2024.

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.

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

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 13358–13376, 2023b.

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, 2024a.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024b.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7057–7075, 2020.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In *The Thirteenth International Conference on Learning Representations*, 2025.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ringattention with blockwise transformers for near-infinite context. In *International Conference on Learning Representations*, volume 2024, pp. 3992–4008, 2024a.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with blockwise ringattention. In *International Conference on Learning Representations*, volume 2025, pp. 45953–45977, 2025.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024b.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Ayan Sengupta, Siddhant Chaudhary, and Tanmoy Chakraborty. Value-guided kv compression for llms via approximated cur decomposition. *Advances in Neural Information Processing Systems*, 38:79461–79492, 2026.

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

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. In *International Conference on Machine Learning*, pp. 47901–47911. PMLR, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 10–19, 2019.

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.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. Infllm: Training-free long-context extrapolation for llms with an efficient context memory. *Advances in neural information processing systems*, 37:119638–119661, 2024a.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024b.

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

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. Xattention: Block sparse attention with antidiagonal scoring. In *Forty-second International Conference on Machine Learning*, 2025.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Haocheng Xi, Jun Zhu, Jianfei Chen, et al. Spargeattention: Accurate and training-free sparse attention accelerating any model inference. In *Forty-second International Conference on Machine Learning*, 2025a.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, et al. bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277, 2024.

Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, Keir Adams, Maurice Weiler, Xiner Li, Tianfan Fu, Yucheng Wang, Haiyang Yu, YuQing Xie, Xiang Fu, Alex Strasser, Shenglong Xu, Yi Liu, Yuanqi Du, Alexandra Saxton, Hongyi Ling, Hannah Lawrence, Hannes Stärk, Shurui Gui, Carl Edwards, Nicholas Gao, Adriana Ladera, Tailin Wu, Elyssa F. Hofgard, Aria Mansouri Tehrani, Rui Wang, Ameya Daigavane, Montgomery Bohde, Jerry Kurtin, Qian Huang, Tuong Phung, Minkai Xu, Chaitanya K. Joshi, Simon V. Mathis, Kamyar Azizzadenesheli, Ada Fang, Alán Aspuru-Guzik, Erik Bekkers, Michael Bronstein, Marinka Zitnik, Anima Anandkumar, Stefano Ermon, Pietro Liò, Rose Yu, Stephan Günnemann, Jure Leskovec, Heng Ji, Jimeng Sun, Regina Barzilay, Tommi Jaakkola, Connor W. Coley, Xiaoning Qian, Xiaofeng Qian, Tess Smidt, and Shuiwang Ji. Artificial

intelligence for science in quantum, atomistic, and continuum systems. *Foundations and Trends® in Machine Learning*, 18(4):385–912, 2025b. ISSN 1935-8237. doi: 10.1561/2200000115. URL http://dx.doi.org/10.1561/2200000115.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

Kan Zhu, Tian Tang, Qinyu Xu, Yile Gu, Zhichen Zeng, Rohan Kadekodi, Liangyu Zhao, Ang Li, Arvind Krishnamurthy, and Baris Kasikci. Tactic: Adaptive sparse attention with clustering and distribution fitting for long-context llms. *arXiv preprint arXiv:2502.12216*, 2025.

A LLM ACCELERATION

LLM inference acceleration. The quadratic cost of self-attention with respect to sequence length has motivated many acceleration methods for long-context LLM inference. One line improves hardware efficiency through IO-aware or distributed attention kernels, including FlashAttention (Dao et al., 2022), FlashDecoding++ (Hong et al., 2024), and RingAttention (Liu et al., 2024a). Another reduces the effective context length through retrieval, prompt compression, or context-memory mechanisms (Borgeaud et al., 2022; Jiang et al., 2023b; 2024b; Xiao et al., 2024a). For decoding, methods address KV-cache efficiency through paged memory management, quantization, selective cache use, and eviction (Kwon et al., 2023; Liu et al., 2024b; Xiao et al., 2025; Zhang et al., 2023). Other approaches change the model architecture through recurrence, state-space models, linear attention, or hybrid designs (Dai et al., 2019; Sun et al., 2023; Gu & Dao, 2023). These approaches reduce inference cost through computation, memory, context, or architectural changes.

Oracle analysis and KV-cache reduction. Gupta & Berant (2021) analyze per-query oracle approximations expressed as sparse convex combinations of value vectors, setting aside computational and memory constraints. VATP (Guo et al., 2024), CAOTE (Goel et al., 2025), and CurDKV (Sengupta et al., 2026) instead address KV-cache reduction, selecting retained states through norm-weighted token importance, eviction-induced output error, and CUR-based scores, respectively. Their selection objects are approximation coefficients or cached KV states. MASA instead ranks framework-defined masks over query–key interactions for structured sparse attention computation.

Sparse attention. Sparse attention reduces computation by retaining only a subset of query–key interactions (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Xiao et al., 2024b). Methods differ in how they construct this subset: fixed local, global, or sink patterns; offline assignment of structured patterns; learned prediction of input-dependent regions; and adaptive selection budgets (Jiang et al., 2024a; Gao et al., 2025; Lai et al., 2025; Zhu et al., 2025). The representative sparse-prefill frameworks reviewed in Sec. 2.1 rank candidate units using attention-weight summaries or learned attention-pattern targets. These criteria estimate where attention is concentrated, rather than directly evaluating the matrix-product error reduction from retaining a candidate unit. Thus, constructing a sparse pattern family and choosing how to rank its units are distinct design questions.

Position and novelty of MASA. MASA is orthogonal to sparse-kernel optimization and KV-cache storage design: **it changes the selection criterion, not the execution kernel or cache representation.** Within sparse attention, MASA also retains the candidate pattern family and budget rule. Pattern construction defines the admissible sparse units, the budget constrains their selection, and the kernel executes the selected interactions. MASA addresses a different question: *which units should be retained to preserve the attention matrix product?*

MASA answers this question through **matrix-approximation ranking of structured sparse units**. The closed-form score in Eq. (7) measures the reduction in retained matrix-product error from selecting each candidate individually. It accounts for alignment with the target product $\boldsymbol{AX}$ and interactions among entries within the unit, yielding a common criterion for vertical, slash, and block patterns. The implementation evaluates this criterion at the framework's existing profile resolution through profile-matched probes and memory-safe matrix products (Sec. 3.3). Raw, pooled, and norm-controlled probes use the same scoring formulation without changing the value vectors consumed by the sparse kernel.

This distinction is the main novelty of MASA: it connects a structured matrix-approximation objective to a common ranking mechanism within existing sparse attention systems. Rather than introducing another sparse pattern, compression policy, or execution kernel, MASA changes what makes a candidate unit important: its contribution to matrix-product preservation, not merely its attention mass. It therefore complements advances in sparse execution and memory management with a theory-grounded correction to the selection principle.

B FROBENIUS NORM AND MATRIX-PRODUCT ERROR

For a matrix $\boldsymbol{Z} \in \mathbb{R}^{m \times n}$, the squared Frobenius norm is

$$\|\boldsymbol{Z}\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \boldsymbol{Z}_{ij}^2 = \mathrm{tr}(\boldsymbol{Z}\boldsymbol{Z}^\top). \tag{12}$$

It measures the total squared error over all matrix entries. In sparse attention, the attention output has one row per query position, so

$$\|\boldsymbol{A}\boldsymbol{V} - (\boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{V}\|_F^2 \tag{13}$$

measures the total squared output error over all query positions and value dimensions.

For a general probe matrix $\boldsymbol{X}$, the MASA loss is

$$\mathcal{L}_{\mathrm{MASA}}(\mathcal{S}; \boldsymbol{X}) = \|(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{X}\|_F^2 . \tag{14}$$

Using Eq. (12), this expands to

$$\mathcal{L}_{\mathrm{MASA}}(\mathcal{S}; \boldsymbol{X}) = \mathrm{tr}\left((\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{X}\boldsymbol{X}^\top\,(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})^\top\right). \tag{15}$$

This is the form used in Sec. 3.2. It shows that MASA does not measure the entrywise error of $\boldsymbol{A}$ alone. It measures the error after the attention matrix acts on $\boldsymbol{X}$.

C Metric View of MASA

The probe matrix $\boldsymbol{X}$ specifies which matrix product should be preserved. The direct choice $\boldsymbol{X} = \boldsymbol{V}$ gives the standard attention-output error:

$$\mathcal{L}_{\mathrm{MASA}}(\mathcal{S}; \boldsymbol{V}) = \|\boldsymbol{A}\boldsymbol{V} - (\boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\boldsymbol{V}\|_F^2 , \tag{16}$$

which is exactly the objective in Eq. (2).

More general output metrics can also be written in the same form. Let $\boldsymbol{H} \succeq 0$ be a positive semidefinite metric over the value dimension, and define

$$\|\boldsymbol{Z}\|_{\boldsymbol{H}}^2 = \mathrm{tr}(\boldsymbol{Z}\boldsymbol{H}\boldsymbol{Z}^\top). \tag{17}$$

Since $\boldsymbol{H} \succeq 0$, there exists a matrix $\boldsymbol{R}$ such that

$$\boldsymbol{H} = \boldsymbol{R}\boldsymbol{R}^\top. \tag{18}$$

Then the weighted attention-output error satisfies

$$\|(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{V}\|_{\boldsymbol{H}}^2 = \mathrm{tr}\left((\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{V}\boldsymbol{H}\boldsymbol{V}^\top\,(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})^\top\right) \tag{19}$$

$$= \mathrm{tr}\left((\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{V}\boldsymbol{R}\boldsymbol{R}^\top\boldsymbol{V}^\top\,(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})^\top\right) \tag{20}$$

$$= \|(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{V}\boldsymbol{R}\|_F^2 . \tag{21}$$

Thus, choosing

$$\boldsymbol{X} = \boldsymbol{V}\boldsymbol{R} \tag{22}$$

turns the weighted output error into the MASA objective:

$$\|(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{V}\|_{\boldsymbol{H}}^2 = \mathcal{L}_{\mathrm{MASA}}(\mathcal{S}; \boldsymbol{V}\boldsymbol{R}). \tag{23}$$

This covers raw value probes, projected value probes, and weighted value-space metrics under the same matrix-product view.

The Frobenius action error also controls the spectral action error. For any matrix $\boldsymbol{Z}$,

$$\|\boldsymbol{Z}\|_{\mathrm{op}} \le \|\boldsymbol{Z}\|_F. \tag{24}$$

Applying this inequality to the MASA residual gives

$$\|(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{X}\|_{\mathrm{op}} \le \|(\boldsymbol{A} - \boldsymbol{M}_{\mathcal{S}} \odot \boldsymbol{A})\,\boldsymbol{X}\|_F . \tag{25}$$

More generally, for Schatten norms,

$$\|\boldsymbol{Z}\|_q \le \|\boldsymbol{Z}\|_p, \qquad 1 \le p \le q \le \infty. \tag{26}$$

Therefore, reducing the Frobenius matrix-product error also reduces an upper bound on stronger matrix-action errors such as the spectral norm. MASA uses the Frobenius objective because it gives the closed-form unit score in Sec. 3.2, while still measuring a matrix-product error rather than an entrywise attention-map error.

D FULL TABLES

The main text reports compact result tables for readability. Here we provide the full comparison layout for each benchmark. Rows marked with † are copied from the corresponding baseline papers and are included only for reference, because they may use different models, sparse budgets, kernels, or evaluation settings. The controlled evidence for MASA comes from the matched rows in each block: the original sparse baseline and the same baseline augmented with MASA. In these matched rows, the model, sparse budget, sparse kernel, sparse unit family, and data processing are kept unchanged; MASA only replaces the sparse-unit ranking score.

Table 5 gives the full RULER comparison across sequence lengths from 4K to 128K. Table 6 gives the full InfiniteBench comparison across summarization, question answering, code, math, and retrieval tasks. Table 7 gives the full LongBench comparison across length buckets. These full tables separate reported prior results from the matched MASA comparison, so the conclusion is not based on mixing results from different settings.

Table 5: RULER (Hsieh et al., 2024) results with additional methods reported in the original baseline papers. Rows marked with † are included for context and may use different settings. In each framework block, Full Attention and reported methods are followed by the matched Original/MASA pair. **Bold** marks the better Avg. within each pair.

		Variant	4K	8K	16K	32K	64K	128K	Avg.
MInfer		Full Attention	97.2	91.8	87.3	80.8	77.4	72.2	84.4
		StreamingLLM†	97.2	38.1	37.5	17.2	14.2	9.4	35.0
		StreamingLLM w/ dilated†	23.4	0.7	1.4	18.8	16.5	15.6	12.7
		StreamingLLM w/ strided†	2.0	0.7	0.6	0.6	0.7	1.3	1.0
		InfLLM†	89.4	79.8	70.1	55.6	43.0	39.5	62.9
		Original	97.7	91.2	88.5	85.0	82.3	77.6	87.0
		MASA	97.93	91.29	88.50	85.45	81.57	78.88	**87.27**
SeerAttn		Full Attention	95.53	92.37	92.01	87.63	84.39	76.26	88.01
		MInference†	95.53	92.64	91.37	85.71	83.24	67.02	85.92
		DuoAttention†	95.64	92.08	90.71	84.75	83.24	75.32	86.96
		Original	95.53	92.71	92.02	88.49	83.48	73.37	87.60
		MASA	95.53	92.74	92.84	89.56	84.01	73.85	**88.09**
FlexPrefill	*Llama*	Full Attention	95.67	93.75	93.03	87.26	84.37	78.13	88.70
		StreamingLLM†	95.43	93.99	74.76	48.56	26.20	30.77	61.62
		MInference†	95.67	93.99	93.27	86.54	84.86	58.17	85.42
		Original	95.43	93.51	94.71	89.42	82.93	79.09	89.18
		MASA	95.43	93.99	94.95	89.66	85.09	80.54	**89.94**
	Qwen	Full Attention	89.90	88.70	80.77	79.33	56.49	17.79	68.83
		StreamingLLM†	90.14	88.94	57.93	43.03	25.48	12.26	52.96
		MInference†	89.90	88.70	79.33	78.61	49.04	10.82	66.07
		Original	90.39	89.91	83.17	81.25	59.14	20.67	70.75
		MASA	90.88	88.94	84.85	82.64	62.79	22.83	**72.16**

Table 6: InfiniteBench (Zhang et al., 2024) results with additional methods reported in the original baseline papers. Rows marked with † are included for context and may use different settings. In each framework block, Full Attention and reported methods are followed by the matched Original/MASA pair. **Bold** marks the better Avg. within each pair.

		Variant	En.Sum	En.QA	En.MC	En.Dia	Zh.QA	Code Debug	Math Find	Retr. PassKey	Retr. Number	Retr. KV	Avg.
MInfer		Full Attention	20.2	12.4	67.3	6.0	12.9	22.1	26.6	100.0	100.0	14.4	38.2
		StreamingLLM†	21.0	8.2	40.2	10.0	10.4	25.9	30.0	86.8	5.1	0.8	23.8
		StreamingLLM w/ dilated†	20.1	9.4	44.5	15.5	11.2	20.5	27.5	5.0	87.5	0.5	24.2
		StreamingLLM w/ strided†	17.3	8.2	27.5	14.5	11.2	19.5	27.5	4.0	2.1	1.0	13.3
		InfLLM†	24.1	7.8	45.0	6.0	11.4	19.5	32.9	100.0	100.0	1.2	34.8
		MInference w/ static†	19.9	8.6	43.2	3.5	8.9	20.6	25.1	92.4	96.3	0.2	31.9
		Original	20.5	12.9	65.9	7.5	12.5	22.3	33.1	100.0	100.0	12.8	38.8
		MASA	19.73	12.15	66.81	8.50	12.28	22.84	33.43	100.00	100.00	16.20	**39.20**
FlexPrefill	*Llama*	Full Attention	31.91	25.92	69.43	21.50	31.95	16.75	24.29	99.15	99.66	60.00	48.06
		StreamingLLM†	30.15	10.15	41.05	8.50	22.38	8.63	17.71	2.71	5.93	0.00	14.72
		MInference†	31.04	22.00	63.76	14.50	28.70	5.33	27.43	56.78	77.12	14.00	34.06
		Original	31.82	24.82	69.43	19.50	35.46	16.75	31.14	98.64	99.83	44.00	47.14
		MASA	32.10	26.25	66.81	19.50	33.66	20.05	29.14	98.31	99.66	54.40	**47.99**
	Qwen	Full Attention	4.65	5.57	34.50	9.00	11.27	24.87	24.57	95.42	75.42	0.00	29.53
		StreamingLLM†	18.54	6.43	39.30	12.50	10.94	24.11	28.00	30.85	61.69	0.00	23.24
		MInference†	7.45	3.94	14.85	4.50	10.17	13.20	30.00	83.90	58.47	0.00	22.65
		Original	14.27	6.55	34.06	13.50	11.51	20.81	30.86	96.95	72.20	0.00	30.07
		MASA	15.55	6.38	32.31	16.50	13.09	21.07	30.29	94.41	74.58	0.00	**30.42**

Table 7: LongBench (Bai et al., 2024b) results with additional methods reported in the original SeerAttention paper. Rows marked with † are included for context and may use different settings. Full Attention and reported methods are followed by the matched Original/MASA pair. **Bold** marks the better Avg. within the pair.

Variant	0–4K	4–8K	8K+	Avg.
Full Attention	55.32	53.98	52.90	54.07
MInference†	55.23	53.78	52.18	53.73
MoA†	50.74	49.84	51.89	50.82
DuoAttention†	53.77	52.17	51.27	52.40
Original	55.43	54.49	52.69	54.20
MASA	55.45	54.48	53.16	**54.37**

E Attention-Output Approximation Error

Direct approximation evidence. We complement the downstream results in Sec. 4.2 by directly measuring how closely the evaluated MASA configurations preserve dense attention outputs. Dense causal FlashAttention-2 and the sparse implementations receive identical $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ tensors. We compare their full-sequence outputs before the attention output projection, including the sparse kernels' softmax normalization.

Error metric. For each evaluated layer, the normalized squared attention-output error is

$$E = \frac{\|\boldsymbol{O}_{\text{sparse}} - \boldsymbol{O}_{\text{dense}}\|_F^2}{\|\boldsymbol{O}_{\text{dense}}\|_F^2}. \tag{27}$$

The Frobenius norm covers the full output matrix. We average errors across the evaluated layers within each prompt and report the aggregate errors. The relative reduction is $100(E_{\text{Original}} - E_{\text{MASA}})/E_{\text{Original}}$; a positive value indicates improved approximation.

Table 8: **Attention-output error decreases across all four evaluated framework–backbone pairs.** Original and MASA denote the baseline and diagnostic configuration. E is the normalized squared output error in Eq. (27). Relative reductions and paired 98.75% confidence intervals are reported in percent. Bold marks the lower error and the positive reduction.

Framework	Backbone	Variant	$E \downarrow$	Reduction (%) $\uparrow$	Paired 98.75% interval (%)
MInference	Llama	Original	1.095985	–	–
		MASA	$\mathbf{1.091532}$	**+0.41**	$[+0.14, +0.70]$
SeerAttention	Llama	Original	1.843×10^{-3}	–	–
		MASA	$\mathbf{1.805 \times 10^{-3}}$	**+2.10**	$[+0.69, +4.44]$
FlexPrefill	Llama	Original	4.9059×10^{-2}	–	–
		MASA	$\mathbf{4.7909 \times 10^{-2}}$	**+2.34**	$[+0.50, +4.18]$
	Qwen	Original	4.2009×10^{-2}	–	–
		MASA	$\mathbf{3.8997 \times 10^{-2}}$	**+7.17**	$[+2.33, +10.82]$

Results. Table 8 shows **lower attention-output error in every evaluated framework–backbone pair, with all four reported paired confidence intervals above zero**. The relative reductions are 0.41% for MInference, 2.10% for SeerAttention, and 2.34% and 7.17% for FlexPrefill on Llama and Qwen, respectively. These measurements directly assess the quantity that motivates MASA: preservation of the attention output, rather than attention mass alone. The evaluated MASA configurations produce outputs closer to dense attention, providing empirical support for the matrix-approximation view alongside the downstream accuracy results.

Evaluation protocol. We use 64K RULER (Hsieh et al., 2024) inputs with a 10% retention target and 208 prompts per model, comprising 16 prompts from each of the 13 task families. Dense and sparse attention receive identical post-RoPE $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ tensors. Errors from eight prespecified layers are averaged within each prompt, and uncertainty is estimated using paired hierarchical task/prompt bootstrap. The reported intervals describe these selected-setting comparisons rather than an independent held-out evaluation. All comparisons are within framework–backbone pairs.

F Full Ablation Tables

Probe configurations. Probe construction determines how value information enters MASA's matrix-product ranking. We compare raw value probes, norm clipping in Eq. (11), and soft norm balancing in Eq. (10), while keeping the sparse baseline, budget, kernel, candidate units, and reduction score in Eq. (7) fixed. The percentile p determines the clipping threshold within the profile region, while α controls the strength of norm balancing.

For each framework, backbone, and benchmark, the main results use the tested probe configuration with the highest aggregate score on that benchmark. The following tables report the evaluated configurations and their full task- and length-level results.

Table 9 reports the RULER ablation across context lengths. Norm control improves the best tested MInference and SeerAttention results, while its effect varies across FlexPrefill configurations. Table 10 reports the InfiniteBench ablation across tasks, comparing raw and norm-controlled probes. Table 11 reports the LongBench ablation for SeerAttention, where clipping at $p = 87.5$ gives the highest average among the tested configurations. Together, these ablations examine probe construction within MASA's matrix-approximation framework, comparing different treatments of value norms under the same closed-form ranking rule and sparse-attention settings.

Table 9: Full RULER probe ablation. Clip p denotes norm clipping at percentile p, and Norm α denotes soft norm balancing. The case $p = -$ and $\alpha = -$ is raw MASA without norm control. **Bold** Avg. marks the best MASA variant within each setting.

Setting	Clip p	Norm α	4K	8K	16K	32K	64K	128K	Avg.
MInference	–	–	97.66	91.46	88.57	85.21	81.76	78.11	87.13
MInference	90	0.6	97.93	91.29	88.50	85.45	81.57	78.88	**87.27**
SeerAttention	–	–	95.53	92.50	92.98	89.58	84.20	72.50	87.88
SeerAttention	85	–	95.63	92.40	91.98	89.55	83.64	73.34	87.76
SeerAttention	90	–	95.53	92.74	92.84	89.56	84.01	73.85	**88.09**
SeerAttention	95	–	95.32	92.84	92.70	89.59	83.52	72.72	87.79
FlexPrefill-Llama	90	–	95.67	94.47	93.99	88.70	84.13	78.61	89.26
FlexPrefill-Llama	95	–	95.67	94.23	93.99	88.94	83.89	78.61	89.22
FlexPrefill-Llama	97.5	–	95.43	93.99	94.95	89.66	85.09	80.54	**89.94**
FlexPrefill-Qwen	–	–	90.88	88.94	84.85	82.64	62.79	22.83	**72.16**
FlexPrefill-Qwen	95	–	90.11	89.94	83.37	82.69	58.44	21.39	70.99
FlexPrefill-Qwen	–	1.0	89.92	88.49	81.73	81.20	59.14	24.03	70.75

Table 10: Full InfiniteBench probe ablation. Clip p denotes norm clipping at percentile p, and Norm α denotes soft norm balancing. The case $p = -$ and $\alpha = -$ is raw MASA without norm control. **Bold** Avg. marks the best MASA variant within each setting.

Setting	Clip p	Norm α	En.Sum	En.QA	En.MC	En.Dia	Zh.QA	Code Debug	Math Find	Retr. PassKey	Retr. Number	Retr. KV	Avg.
MInference	–	–	20.51	13.14	66.38	9.00	12.34	22.34	32.57	100.00	100.00	14.20	39.05
MInference	90	0.6	21.13	13.02	66.38	8.00	12.31	23.10	32.00	100.00	100.00	14.80	39.07
MInference	80	0.7	19.73	12.15	66.81	8.50	12.28	22.84	33.43	100.00	100.00	16.20	**39.20**
FlexPrefill-Llama	–	0.5	32.10	26.25	66.81	19.50	33.66	20.05	29.14	98.31	99.66	54.40	**47.99**
FlexPrefill-Qwen	–	–	15.55	6.38	32.31	16.50	13.09	21.07	30.29	94.41	74.58	0.00	**30.42**
FlexPrefill-Qwen	–	0.5	15.65	5.98	34.50	14.50	11.77	19.29	33.43	94.07	73.90	0.00	30.31

Table 11: Full LongBench probe ablation for SeerAttention. Clip p denotes norm clipping at percentile p. The case $p = -$ and $\alpha = -$ is raw MASA without norm control. **Bold** Avg. marks the best MASA variant.

Clip p	Norm α	0–4K	4–8K	8K+	Avg.
–	–	55.31	54.57	52.84	54.24
85	–	55.27	54.79	52.70	54.25
87.5	–	55.45	54.48	53.16	**54.37**
90	–	55.27	54.65	52.92	54.28