# BLOCK-LEVEL POSITIONAL RECONSTRUCTION FOR SPARSE LONG-CONTEXT PREFILLING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Long-context inference has become a defining capability of large language models, but exact dense attention remains costly due to its quadratic scaling with sequence length. Block-sparse attention offers a hardware-friendly alternative by selecting a small set of relevant key blocks for each query block, yet estimating block importance accurately without model retraining or costly token-level search remains challenging. Existing block estimators often pool RoPE-rotated token representations, causing within-block phase variation to induce frequency-dependent cancellation and distort position-sensitive block affinities. We empirically find that, although unrotated block summaries cannot exactly recover the omitted phase-aware interaction, its predictable component follows a frequency- and distance-conditioned response that transfers across input sequences. Based on this observation, we propose **RePhase**, which combines current unrotated block interactions with a reusable positional-response profile and the exact relative RoPE phase. The profile is fitted once from unlabeled forward-pass statistics without backpropagation or model-parameter updates, and enables direct block-level scoring at inference without token-level search. Experiments on long-context text and video tasks show that RePhase approaches full-attention performance across 4K–128K contexts, keeps block-estimation overhead below 3.4 ms, and achieves a 5.03× speedup over FlashAttention at 128K.

## 1 INTRODUCTION

Long-context inference has become a core capability of large language models (LLMs) Tang et al. (2024); Zeng et al. (2024); Yang et al. (2025); Team et al. (2026); Mao et al. (2026); Tang et al. (2026); Jiang et al. (2026), as they move from short isolated prompts toward tasks that require retrieving and integrating information from long documents Duan et al. (2025); Hu et al. (2024), multi-turn conversations Bai et al. (2025), and lengthy videos Grauman et al. (2024). In Transformer-based LLMs, this integration is largely mediated by self-attention, which models token-to-token interactions across the context. However, standard dense self-attention computes pairwise interactions among all tokens, making its cost grow quadratically with sequence length. Sparse attention has been widely adopted for long-context prefilling to reduce this cost by selecting only a subset of context positions for each query Yuan et al. (2025); Wang et al. (2025); Xu et al. (2026); Zhou et al. (2026); Bai et al. (2026). However, identifying that subset through token-level search can itself be expensive. As contexts grow longer, this search can erode the savings of sparse computation and sometimes make sparse attention slower than optimized dense attention. Block-sparse attention Xu et al. (2025); Wu et al. (2025) mitigates this issue by grouping contiguous tokens into local blocks, which better matches the tiling and matrix-computation patterns of modern GPUs. Its inference pipeline has two stages: estimating key-block importance for each query block and computing sparse attention over the selected block pairs. The key challenge is therefore to make block estimation both accurate and lightweight: poor estimates discard important context, whereas expensive estimators erode the efficiency gains of sparsity.

Existing block estimators commonly pool RoPE-rotated token representations to approximate block importance Lai et al. (2025); Wang et al. (2026). Although efficient, this design applies position-dependent rotation before aggregating block content. At higher RoPE frequencies, the rotation phase changes more rapidly across token positions within a block. Pooling these rotated representations can therefore induce destructive phase interference, attenuating high-frequency components and



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.

weakening the local positional structure available for block estimation, as illustrated in Figure 6. Recent spectral-aware work mitigates this attenuation through frequency-wise scoring and energy-based calibration Wang et al. (2026), but still derives block affinities from representations pooled after RoPE. This leaves open whether block content can instead be aggregated in the unrotated space and its positional response reconstructed directly at the block level.

Our key observation is that the positional response omitted by unrotated block aggregation is not input-specific noise: it exhibits stable patterns across rotary frequencies, relative block displacements, and input sequences. Based on this observation, we propose **RePhase**, a block estimator that combines input-specific block content with reusable positional structure. Specifically, RePhase first aggregates queries and keys in the unrotated space, and then combines their block interactions with a frequency- and distance-dependent positional-response profile and relative RoPE phases to compute closed-form block importance scores. The profile is estimated once from forward-pass statistics over a small, unlabeled calibration corpus, without backpropagation or model-parameter updates, and is reused across inference inputs. Because all input-dependent scoring operates on block summaries, RePhase avoids token-level search and keeps block-estimation overhead low. We further implement RePhase with a custom Triton kernel to execute block estimation efficiently on GPUs.

We evaluate RePhase on long-context retrieval, understanding, language modeling, and video reasoning across dense, mixture-of-experts, hybrid, and vision-language backbones. Across context lengths from 4K to 128K, RePhase consistently approaches full-attention performance and achieves the best or competitive results among strong sparse-attention baselines. Under matched block budgets, RePhase retains the most dense-attention mass and achieves the highest Top-$K$ recall at every evaluated context length, indicating that its gains come from better block ranking rather than selecting more blocks. Its block-estimation latency remains below 3.4 ms across all evaluated sequence lengths, contributing to an overall 5.03$\times$ prefill speedup over FlashAttention at 128K. Our main contributions are summarized as follows:

**(1) Block-Level Positional Response.** We decompose RoPE-rotated block interactions into relative-block and within-block components, and show that the within-block response exhibits frequency-dependent patterns shared across inputs.

**(2) RePhase.** We introduce **RePhase**, which combines unrotated block interactions, a positional-response profile estimated from unlabeled forward passes, and relative RoPE phases to score blocks without backpropagation, parameter updates, or token-level search.

**(3) Accuracy and Efficiency.** Experiments show that RePhase approaches full-attention performance while improving the accuracy–efficiency trade-off over existing sparse-attention methods.

## 2 Background and Related Work

Sparse attention has been widely adopted for long-context prefilling to reduce the quadratic cost of self-attention by restricting computation to selected query–key interactions Nawrot et al. (2025); Hu et al. (2026); Desai et al. (2025); Xiu et al. (2025). Existing methods can be broadly categorized into native and inference-time sparse attention methods. Native approaches learn sparse attention jointly with model parameters, but require substantial training compute and data Yuan et al. (2025); Liu et al. (2025); Lu et al. (2026); Gao et al. (2024). Inference-time methods instead derive sparse patterns from forward-pass activations without modifying model parameters, which makes them directly applicable to pretrained full-attention checkpoints Mazaré et al. (2025); Zhang et al. (2025).

Beyond this distinction, sparse attention methods also differ in their selection granularity. Existing methods primarily operate at either token or block granularity Nawrot et al. (2025); Jo et al. (2026); Liu et al. (2026). Token-level methods estimate relevance at individual token positions and preserve the fine-grained information needed for selection Desai et al. (2025). However, token-wise scoring and non-contiguous indexing can introduce irregular memory access and erode the gains of sparse computation. Block-sparse attention instead groups contiguous tokens into fixed-size blocks and computes only selected block pairs Lai et al. (2025); Xu et al. (2025); Wu et al. (2025). This regular structure aligns with the tiled execution of modern GPU attention kernels and improves hardware utilization Gao et al. (2025). To keep selection inexpensive, block-level estimators commonly compress each block into a compact representation. This compression reduces estimation cost but can obscure fine-grained relevance, making accurate block estimation challenging.

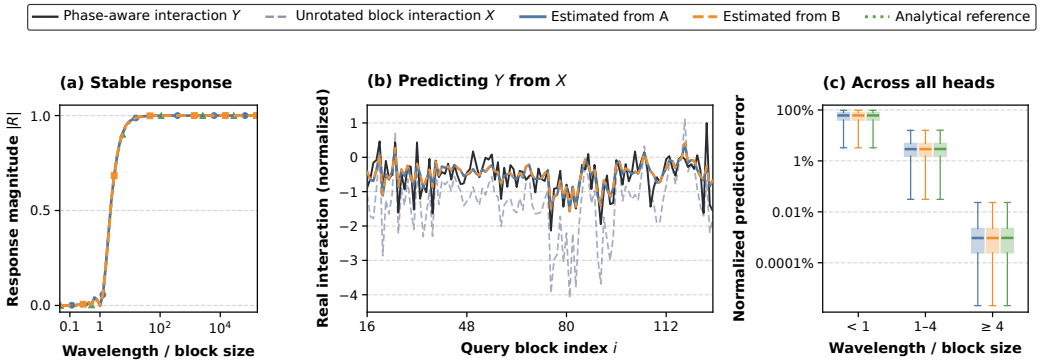



Figure 1: **Stable positional responses transfer across inputs.** On Qwen3-8B, we estimate empirical response coefficients $R$ from disjoint text subsets A and B and evaluate them on held-out sequences. Here, $X$ is the unrotated block interaction and $Y$ its phase-aware counterpart. (a) Independently estimated coefficient magnitudes exhibit closely aligned frequency patterns. (b) On unseen input, predictions $RX$ more closely follow the measured $Y$ than $X$ alone does. (c) Across layers and heads, prediction errors are lower at longer wavelengths.

A common block estimator summarizes each block by averaging its RoPE-rotated queries and keys Su et al. (2023); Wang et al. (2026). It scores each query–key block pair from the affinity between the resulting summaries, avoiding token-pair search while retaining efficient block-level matrix multiplication. However, RoPE applies position-dependent rotations before pooling, causing representations with different within-block phases to be aggregated together. This produces frequency-dependent attenuation: rapidly rotating components can cancel during pooling, whereas slowly rotating components are largely preserved. Recent work compensates for this attenuation through frequency-aware scoring and calibration Wang et al. (2026). Such corrections retain block-level efficiency but still operate on representations pooled after RoPE. RePhase instead aggregates current block content in the unrotated space and reconstructs positional structure at the block level.

## 3 Methodology

RePhase estimates block importance from input-specific block content and a reusable profile of block-level positional response. We first use the RoPE-rotated pooled interaction as an offline phase-aware reference and decompose it into the known phase between block positions and a within-block phase-aware interaction. We then relate the latter to the corresponding unrotated block interaction. Their predictable relationship is conditioned on rotary frequency and relative block displacement, yet remains stable across inputs. RePhase estimates this transferable response once from forward-pass statistics and combines it with the current unrotated block interaction and the relative phase for online scoring.

### 3.1 Reconstruction Objective

**Phase-Aware Reference Interaction.** Consider an off-diagonal causal pair formed by query block $i$ and key block $j < i$. Let $\Delta = i - j$ denote their relative block displacement, and let $n, m \in \{0, \dots, B-1\}$ denote the token offsets inside the two blocks. For a head of dimension $d$, let $f$ index its $d/2$ complex rotary subspaces, and let $\omega_f$ denote the corresponding rotary frequency. Let $q_{r,f}, k_{r,f} \in \mathbb{C}$ denote the unrotated query and key at position $r$, and use the superscript R for their representations after RoPE. We define the phase-aware reference as the mean interaction over all $B^2$ token pairs after RoPE. Each rotary subspace in complex form factorizes this reference exactly into

the relative phase between the two block positions and a within-block phase-aware interaction:

$$S^{\mathrm{ref}}(i,j) = \frac{1}{B^2\sqrt{d}} \sum_{n,m=0}^{B-1} \langle q^{\mathrm{R}}_{iB+n}, k^{\mathrm{R}}_{jB+m} \rangle = \frac{1}{\sqrt{d}} \sum_{f=1}^{d/2} \mathrm{Re}\big[Y_f(i,j) e^{\mathrm{i}\omega_f B\Delta}\big] ,$$
$$\text{where} \quad Y_f(i,j) = \frac{1}{B^2} \sum_{n,m=0}^{B-1} q_{iB+n,f} \big(k_{jB+m,f}\big)^* e^{\mathrm{i}\omega_f (n-m)}. \tag{1}$$

The relative block phase $e^{\mathrm{i}\omega_f B\Delta}$ is determined entirely by $\Delta$ and can be evaluated analytically. The remaining term $Y_f(i,j)$ combines the current token content with the RoPE phases induced by its offsets inside the two blocks. This factorization separates the known relative block phase from the within-block positional response used for profile estimation. See Appendix B.1 for the derivation.

**Unrotated block interaction.** To represent the current block content without mixing different RoPE phases, we average the unrotated queries and keys within each block:

$$\bar{q}_{i,f} = \frac{1}{B} \sum_{n=0}^{B-1} q_{iB+n,f}, \qquad \bar{k}_{j,f} = \frac{1}{B} \sum_{m=0}^{B-1} k_{jB+m,f}, \qquad X_f(i,j) = \bar{q}_{i,f} \big(\bar{k}_{j,f}\big)^*. \tag{2}$$

The resulting interaction $X_f(i,j)$ describes the content of the current query–key block pair but contains no explicit within-block phase. In contrast, $Y_f(i,j)$ also depends on how that content is distributed across token offsets inside the two blocks. Recovering this interaction $Y_f(i,j)$ from $X_f(i,j)$ would require a separate content-dependent response for every query–key offset pair in every rotary subspace. Such fine-grained positional responses cannot be represented by a single analytical factor at the block level. We therefore turn to their aggregate behavior and examine whether it exhibits a stable block-level positional preference across inputs.

### 3.2 Stable Block-Level Positional Responses

Although individual interactions vary across inputs, the aggregate relationship between $X_f(i,j)$ and $Y_f(i,j)$ exhibits stable block-level structure. We characterize this relationship across rotary frequencies and relative block displacements, and then test whether it transfers across input sequences.

**Empirical positional response.** For each rotary frequency and relative block displacement, we fit a complex response coefficient that maps the unrotated interaction $X_f(i,j)$ to the phase-aware interaction $Y_f(i,j)$. For an input set $\mathcal{D}$, let $\mathcal{P}_{\mathcal{D},\Delta}(s,i,j) \mid s \in \mathcal{D},\ i-j=\Delta$ collect its block pairs at displacement $\Delta$, and let $M_{\mathcal{D},\Delta} = |\mathcal{P}_{\mathcal{D},\Delta}|$. Restoring the layer and head indices, the empirical response coefficient is

$$R^{\mathcal{D}}_{\ell,h,f}(\Delta) = \frac{\sum_{(s,i,j)\in\mathcal{P}_{\mathcal{D},\Delta}} Y_{s,\ell,h,f}(i,j) X_{s,\ell,h,f}(i,j)^*}{\sum_{(s,i,j)\in\mathcal{P}_{\mathcal{D},\Delta}} |X_{s,\ell,h,f}(i,j)|^2 + M_{\mathcal{D},\Delta}\lambda}. \tag{3}$$

Eq. 3 gives the closed-form response coefficient that best relates $X$ to $Y$ under regularized complex fitting. Its magnitude measures the amplitude modulation applied to the unrotated block interaction, while its angle captures the corresponding phase shift. Because the relative block phase has already been factored out of $Y_f(i,j)$ in Equation 1, $R^{\mathcal{D}}_{\ell,h,f}(\Delta)$ captures only the within-block positional response. See Appendix B.2 for the derivation.

**Empirical validation.** We estimate empirical response coefficients independently from two disjoint input subsets and evaluate the resulting predictions on a common set of held-out sequences. Figure 1 reveals three complementary observations. First, response magnitudes estimated from different inputs exhibit closely aligned frequency-dependent patterns (a). Second, the estimated coefficients transfer to new content: multiplying the current unrotated interaction $X_f(i,j)$ by either estimate produces predictions closer to the measured phase-aware interaction $Y_f(i,j)$ than $X_f(i,j)$ alone on an unseen sequence (b). Third, held-out error distributions across all layers and heads show lower reconstruction errors in longer-wavelength bands (c).

Together, these observations support approximating the phase-aware interaction $Y_f(i,j)$ from input-specific block content and a shared within-block positional response. This motivates RePhase's

offline–online decomposition: estimate a reusable positional profile once from calibration data, then combine it with the current unrotated block interaction to reconstruct the predictable component of $Y_f(i,j)$ during inference.

### 3.3 REPHASE: BLOCK-LEVEL POSITIONAL RECONSTRUCTION

Based on the analysis in Section 3.2, we propose RePhase, a block estimator that reconstructs positional structure directly at the block level. RePhase combines the unrotated interaction $X_{s,\ell,h,f}(i,j)$, which retains the input-specific interaction between the current query and key blocks, with a reusable positional profile $\hat{\mathcal{C}}_{\ell,h,f}(\Delta)$, which captures the stable preference conditioned on rotary frequency and relative block displacement. For a block pair at displacement $\Delta$, RePhase models their relationship as $Y_{s,\ell,h,f}(i,j) \approx \hat{\mathcal{C}}_{\ell,h,f}(\Delta) X_{s,\ell,h,f}(i,j)$. We next describe how the positional profile is estimated offline and used for online block scoring.

**Offline positional profiling.** Given a small, unlabeled, and task-agnostic calibration corpus $\mathcal{D}_{\text{cal}}$, we compute the positional profile as $\hat{\mathcal{C}}_{\ell,h,f}(\Delta) = R^{\mathcal{D}_{\text{cal}}}_{\ell,h,f}(\Delta)$ using Equation 3. This procedure uses only forward-pass statistics and requires neither backpropagation nor model-parameter updates. For a fixed model, RoPE configuration, and block size, profiling is performed once, and the resulting coefficients are reused across inference inputs. The content of each new input is supplied online through $X_{s,\ell,h,f}(i,j)$.

**Online block scoring.** At inference, RePhase computes the input-specific interaction $X_f(i,j)$ from the unrotated block summaries of the current sequence. It then combines this interaction with the offline positional profile and the exact relative phase between the two block positions. Omitting the sequence, layer, and head indices for clarity, substituting the profiled approximation into Equation 1 gives

$$S_{\text{RePhase}}(i,j) = \frac{1}{\sqrt{d}} \sum_{f=1}^{d/2} \text{Re}\left[\hat{\mathcal{C}}_f(\Delta) X_f(i,j) e^{\text{i}\omega_f B\Delta}\right], \qquad \Delta = i - j. \tag{4}$$

Here, $X_f(i,j)$ supplies the interaction between the current query and key blocks, $\hat{\mathcal{C}}_f(\Delta)$ supplies their shared within-block positional response, and $e^{\text{i}\omega_f B\Delta}$ restores their exact relative phase. The complex profile admits the polar form $\hat{\mathcal{C}}_f(\Delta) = \hat{\Lambda}_f(\Delta) e^{\text{i}\hat{\theta}_f(\Delta)}$, where $\hat{\Lambda}_f(\Delta)$ and $\hat{\theta}_f(\Delta)$ denote its amplitude and phase responses, respectively. These components correspond to the amplitude-only and phase-only ablations. Appendix B.3 provides the equivalent real-valued form of Equation 4. Because every input-dependent quantity is computed from block summaries, RePhase scores block pairs without token-level search. The calibration corpus covers relative block displacements only up to $\Delta_{\max}$. For $\Delta > \Delta_{\max}$, RePhase uses $\hat{\mathcal{C}}_f(\Delta_{\max})$ for the profile lookup while evaluating $e^{\text{i}\omega_f B\Delta}$ at the actual displacement $\Delta$. Thus, only the profiled coefficient is clamped; the analytical RoPE phase continues to reflect the true distance between the two blocks.

**Sparse selection and complexity.** For each query block, RePhase applies the causal block mask, normalizes the valid scores with a softmax, and performs deterministic top-$p$ selection. The selection retains the smallest set of highest-probability key blocks whose cumulative probability reaches $p$. We add the sink and diagonal blocks if they are not selected and preserve the token-level causal mask within every retained block pair. The resulting mask is passed to a block-sparse kernel for exact token-level attention. Let $N = \lceil L/B \rceil$ be the number of blocks and let $\mathcal{M}$ denote the selected block pairs. Constructing the block summaries costs $\mathcal{O}(Ld)$, scoring all causal block pairs costs $\mathcal{O}(N^2 d)$, and the subsequent sparse attention costs $\mathcal{O}(|\mathcal{M}|B^2 d)$. The positional profile adds only indexed coefficient lookups and does not change these asymptotic costs.

## 4 EXPERIMENTS

### 4.1 EXPERIMENTAL SETTINGS

We evaluate RePhase on a range of long-context text tasks covering retrieval, understanding, reasoning, and language modeling. For controlled needle-in-a-haystack retrieval and aggregation, we use RULER Hsieh et al. (2024). To evaluate deep understanding and multi-hop reasoning over long contexts, we use LongBench Bai et al. (2024) and LongBench v2 Bai et al. (2025). To examine

| | *Llama-3.1-8B-Instruct* | | | | | | | *Qwen3-8B with YaRN* | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **4K** | **8K** | **16K** | **32K** | **64K** | **128K** | **Avg.** | **4K** | **8K** | **16K** | **32K** | **64K** | **128K** | **Avg.** |
| FlashAttention | 95.42 | 94.38 | 93.38 | 87.98 | 84.72 | 77.77 | 88.94 | 95.01 | 92.35 | 90.04 | 87.24 | 79.93 | 75.09 | 86.61 |
| StreamingLLM | 95.34 | 67.86 | 54.29 | 39.02 | 21.53 | 21.01 | 49.84 | 93.54 | 76.51 | 61.09 | 54.79 | 43.78 | 38.78 | 61.41 |
| MInference | **95.43** | 94.46 | **93.42** | 87.22 | 83.07 | 71.04 | 87.44 | **95.08** | **92.37** | **89.67** | 86.01 | 76.53 | 70.36 | 85.00 |
| FlexPrefill | 93.80 | 92.44 | 93.28 | 87.92 | **84.74** | 72.41 | 87.43 | 90.89 | 87.61 | 87.82 | 85.58 | 78.27 | 73.42 | 83.93 |
| XAttention | 95.17 | 94.30 | 93.28 | **89.06** | 82.31 | 70.52 | 87.44 | 94.55 | 91.03 | 87.91 | 84.37 | 77.73 | 72.01 | 84.60 |
| Prism | 95.28 | **94.47** | 92.48 | 87.67 | 82.59 | 72.75 | 87.54 | 94.84 | 90.95 | 87.69 | 86.88 | 78.58 | 72.65 | 85.27 |
| **RePhase (Ours)** | 95.35 | 94.26 | 92.74 | 86.54 | 82.82 | **73.57** | **87.55** | 94.97 | 92.05 | 88.72 | **87.67** | **79.02** | **74.11** | **86.09** |

Table 1: RULER accuracy across context lengths. Bold and underlined values denote the best and second-best results among sparse-attention methods, respectively.

| | *Llama-3.1-8B-Instruct* | | | | | | | *Qwen3-8B with YaRN* | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **S-Doc** | **M-Doc** | **Summ.** | **Few** | **Code** | **Synth.** | **Avg.** | **S-Doc** | **M-Doc** | **Summ.** | **Few** | **Code** | **Synth.** | **Avg.** |
| FlashAttention | 47.51 | 43.28 | 25.90 | 45.92 | 18.01 | 68.18 | 41.47 | 47.10 | 40.45 | 24.07 | 56.69 | 1.65 | 67.00 | 39.49 |
| StreamingLLM | 34.89 | 27.47 | 24.81 | **46.67** | 17.18 | 26.24 | 29.54 | 40.27 | 30.64 | 23.23 | 49.82 | 0.91 | 36.67 | 30.26 |
| MInference | **47.42** | 42.54 | 25.85 | 45.58 | 17.84 | **67.60** | **41.14** | **46.90** | **40.39** | 24.07 | 55.74 | 1.61 | **66.33** | **39.18** |
| FlexPrefill | 46.13 | 41.49 | 25.85 | 46.63 | 17.68 | 25.61 | 33.90 | 43.77 | 39.31 | 23.99 | 57.33 | **1.87** | 50.50 | 36.13 |
| XAttention | 45.89 | 41.56 | **26.18** | 45.86 | **19.24** | 59.32 | 39.68 | 44.49 | 40.09 | **24.12** | 57.27 | 1.29 | 65.67 | 38.82 |
| Prism | 47.09 | 42.13 | 26.00 | 46.40 | 18.72 | 66.15 | 41.08 | 46.47 | 40.08 | 24.01 | 58.36 | 1.64 | 64.17 | 39.12 |
| **RePhase (Ours)** | 45.27 | **42.61** | 26.03 | 46.24 | 18.63 | 65.60 | 40.73 | 45.11 | 40.15 | **24.12** | **58.60** | 1.71 | 65.00 | 39.12 |

Table 2: LongBench scores across task categories. S-Doc, M-Doc, Few, and Synth. denote single-document QA, multi-document QA, few-shot learning, and synthetic tasks. MInference achieves the best result in several categories, but below 32K—the regime covering LongBench—it may lose its sparsity advantage and become slower than full attention. RePhase achieves comparable accuracy with lower prefill latency (Figure 2).

whether sparsification preserves long-form text modeling fidelity, we report perplexity on PG19 Rae et al. (2019). We also test RePhase on long-video reasoning using VideoMME Fu et al. (2025) and LongVideoBench Wu et al. (2024). Due to space constraints, we defer descriptions of the evaluated models and baselines, together with further implementation details, to Appendix A.1.

### 4.2 Main Results

**Long-Context Retrieval.** RePhase remains accurate as the retrieval space grows (Table 1). It achieves the highest average score among sparse methods on both Llama-3.1-8B-Instruct and Qwen3-8B, reaching 87.55 and 86.09, respectively. These results remain within 1.39 and 0.52 points of full attention. The advantage becomes clearer at 128K, where RePhase reaches 73.57 on Llama and 74.11 on Qwen, outperforming the strongest sparse baselines by 0.82 and 0.69 points despite using only block-level features. At shorter contexts, competitive estimators remain close because each query block selects from a smaller candidate set. As the context grows, estimation errors increasingly alter the selected blocks and lead to larger accuracy drops. RePhase degrades more slowly because it reconstructs positional structure before block selection, making block-level estimation more reliable at long context lengths.

**Long-Context Understanding.** LongBench tests diverse attention patterns beyond retrieval (Table 2). No sparse method dominates every category because different tasks require different forms of context aggregation. RePhase performs particularly well on multi-document QA with Llama and few-shot learning with Qwen. These tasks rely on different evidence structures, showing that its block estimator is not specialized to a single retrieval pattern. LongBench-v2 further varies reasoning difficulty and context span (Table 3). RePhase achieves the best or tied-best sparse average on both backbones. Its clearest advantage appears on long Llama examples, where it surpasses full attention by 2.7 points. Longer inputs introduce more distracting context. This gain suggests that accurate block selection can suppress irrelevant information rather than merely approximate dense attention. Together, these results show that RePhase remains effective as the task structure, difficulty, and context span change.

| Method | *Llama-3.1-8B-Instruct* | | | | | | *Qwen3-8B with YaRN* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | E | H | S<32K | M<128K | L>128K | Overall | E | H | S<32K | M<128K | L>128K | Overall |
| FlashAttention | 31.8 | 28.3 | 33.9 | 27.4 | 26.9 | 29.6 | 38.0 | 33.4 | 40.6 | 32.1 | 32.4 | 35.2 |
| StreamingLLM | 26.0 | 27.3 | **35.6** | 22.3 | 21.3 | 26.8 | 29.7 | 31.2 | 37.2 | 27.4 | 25.9 | 30.6 |
| MInference | **33.9** | 26.7 | 34.4 | 26.0 | 27.8 | 29.4 | 29.7 | 28.6 | 36.1 | 23.7 | 27.8 | 29.0 |
| FlexPrefill | 28.1 | 24.1 | 25.6 | 23.3 | **30.6** | 25.6 | 34.4 | 30.5 | 36.1 | **30.7** | 27.8 | 32.0 |
| XAttention | 32.8 | 26.7 | 34.4 | 24.2 | 29.6 | 29.0 | 34.4 | **32.8** | 40.6 | 29.8 | 27.8 | **32.8** |
| Prism | **33.9** | 26.7 | 33.9 | **27.0** | **30.6** | 29.8 | **36.5** | 29.9 | 37.8 | 29.8 | **28.7** | 32.4 |
| **RePhase (Ours)** | **33.9** | **27.7** | 33.9 | **27.0** | 29.6 | **30.0** | **36.5** | 30.5 | **41.1** | 28.4 | 27.8 | **32.8** |

Table 3: LongBench-v2 accuracy across difficulty and context length. E and H denote Easy and Hard, while S, M, and L denote Short, Medium, and Long contexts. Lengths are measured in words.

| Method | Context Length | | | | | | |
|---|---|---|---|---|---|---|---|
| | 4K | 8K | 16K | 32K | 64K | 128K | Avg. |
| FlashAttention | 27.01 | 25.84 | 25.19 | 24.81 | 24.42 | 24.22 | 25.25 |
| StreamingLLM | 28.77 | 29.07 | 29.55 | 30.09 | 30.71 | 33.37 | 30.26 |
| MInference | 27.02 | 25.93 | 25.44 | 25.30 | 25.18 | 25.93 | 25.80 |
| FlexPrefill | 29.63 | 28.25 | 27.33 | 26.98 | 26.44 | 26.04 | 27.45 |
| XAttention | 27.02 | 25.79 | 25.04 | 24.56 | 24.15 | 23.97 | 25.09 |
| Prism | 27.10 | 25.89 | 25.18 | 24.73 | 24.37 | 24.23 | 25.25 |
| **RePhase (Ours)** | 27.04 | 25.85 | 25.17 | 24.75 | 24.37 | 24.23 | 25.23 |

Table 4: Long-text modeling fidelity on PG19 across context lengths. RePhase consistently matches full attention and remains close to the best-performing method, showing that its selected blocks preserve the predictive information required for prefilling. Lower PPL is better.

| Method | VideoMME | | | | LVB |
|---|---|---|---|---|---|
| | Short | Med. | Long | Overall | Overall |
| FlashAttention | 79.89 | 70.67 | 63.11 | 71.22 | 65.00 |
| StreamingLLM | 64.33 | 50.11 | 50.67 | 55.03 | 50.93 |
| MInference | **79.44** | 70.00 | 62.44 | 70.63 | 61.48 |
| FlexPrefill | 77.67 | **70.67** | 62.67 | 70.34 | 64.10 |
| XAttention | 79.22 | 69.78 | 63.44 | 70.81 | 64.25 |
| Prism | 79.00 | **70.67** | **64.00** | **71.22** | 64.25 |
| **RePhase** | 79.00 | 70.56 | 63.00 | 70.85 | **64.96** |

Table 5: Accuracy on VideoMME and LongVideoBench (LVB). Higher is better. Bold and underlined values denote the best and second-best sparse-attention results.

**Long-Text Modeling Fidelity.** Using Llama-3.1-8B-Instruct, RePhase recovers the predictive fidelity of full attention under sparse prefilling (Table 4). Across all context lengths, its perplexity remains within 0.06 of full attention and within 0.26 of the best-performing method. RePhase further surpasses full attention at 16K, 32K, and 64K, while achieving a comparable average perplexity of 25.23 versus 25.25. The consistently small gap shows that RePhase identifies the blocks carrying the most important predictive information. As a result, sparse attention preserves the long-text modeling behavior of dense attention even as the context grows to 128K.

**Long-Video Reasoning.** RePhase extends its block-level estimation to multimodal long contexts (Table 5). It reaches 70.85 on VideoMME, within 0.37 points of full attention, with consistently small gaps across short, medium, and long videos. On LongVideoBench, RePhase achieves the best sparse-attention result of 64.96, nearly matching the full-attention score of 65.00. These results show that block-level positional reconstruction remains effective under interleaved M-RoPE and preserves temporally distributed evidence beyond text-only inputs.

### 4.3 Efficiency Results

**Prefill Latency and Speedup.** RePhase makes sparse prefilling efficient across the entire context range (Figure 2). It achieves the lowest latency at every evaluated length, while its speedup over FlashAttention grows steadily from 1.29× at 8K to 5.03× at 128K. In contrast, MInference remains slower than FlashAttention from 8K to 32K because its overhead exceeds the computation saved by sparsity. It becomes advantageous only at longer contexts and approaches RePhase at 128K, reaching 94.21 ms versus 89.59 ms. XAttention achieves strong task performance at moderate context lengths, but its latency grows to 320.09 ms at 128K, 257.3% higher than RePhase. RePhase therefore shifts the effectiveness–efficiency Pareto frontier across context scales: it avoids the short-context slowdown of MInference and the long-context overhead of XAttention, while retaining stronger long-context effectiveness.

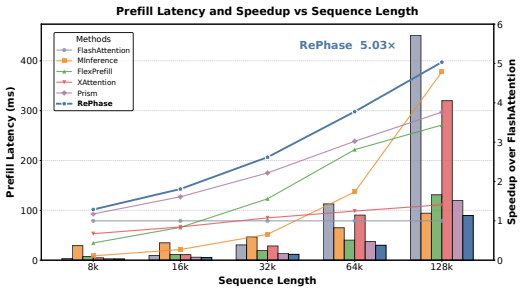



Figure 2: Attention-prefill latency and speedup over FlashAttention across sequence lengths. RePhase achieves the lowest latency across all settings and reaches a 5.03× speedup at 128K.

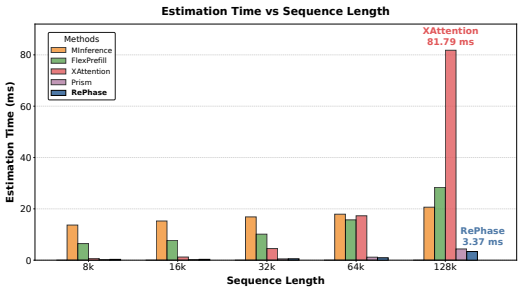



Figure 3: Block-estimation latency across sequence lengths. RePhase keeps estimation overhead below 3.4 ms from 8K to 128K.

| Variant | 4K | 16K | 64K | 128K | Avg. |
|---|---|---|---|---|---|
| **RePhase** | **94.97** | **88.72** | **79.02** | **74.11** | **84.21** |
| RoPE-rotated mean pooling | 94.09 | 86.57 | 77.88 | 71.28 | 82.45 |
| Unrotated block interaction | 91.27 | 84.55 | 77.30 | 70.53 | 80.91 |
| Unrotated interaction + analytical phase | 87.25 | 81.19 | 70.38 | 67.81 | 76.66 |
| Analytical phase + amplitude-only profile | 94.08 | 86.58 | 78.44 | 70.85 | 82.49 |
| Analytical phase + phase-only profile | 87.15 | 81.40 | 72.06 | 67.44 | 77.01 |

Table 6: Component ablation on Qwen3-8B with YaRN. All values are RULER accuracy (%). The amplitude-only and phase-only profiles independently augment the analytical relative phase.

| Calibration Corpus | RULER ↑ | PG19 PPL ↓ |
|---|---|---|
| Paul Graham | 74.11 | 33.172 |
| FineWeb | 74.44 | 33.168 |
| Wikipedia | 74.10 | **33.163** |
| peS2o | **74.50** | 33.173 |

Table 7: Cross-corpus generalization of the offline positional profile at 128K. Profiles calibrated on different corpora maintain comparable RULER accuracy and PG19 perplexity (w/ Qwen3-8B).

**Block-Estimation Overhead.** The prefill scaling difference originates in the estimator (Figure 3). MInference introduces substantial overhead at short contexts, whereas the token-level search used by XAttention and FlexPrefill grows rapidly with sequence length. RePhase instead performs input-dependent computation only after block aggregation and reuses its offline structural profile. Its estimation cost therefore remains nearly constant as the context grows. This allows the computation saved by sparsity to translate directly into attention-prefill latency/speedup

### 4.4 Case Studies

**Component Ablation.** Table 6 examines the components of the RePhase block score. Full RePhase consistently outperforms RoPE-rotated pooling, with the largest advantage at 128K. The content-only and analytical-phase variants both remain below the full model. Avoiding phase cancellation and restoring the relative block phase are therefore insufficient to recover the within-block positional response. The amplitude-only profile recovers most of the lost performance, whereas the phase-only profile remains insufficient. The full complex profile performs best by combining both corrections. These results show that amplitude calibration provides the primary correction, while the residual phase provides a complementary gain.

**Calibration-Corpus Ablation.** Table 7 changes only the calibration corpus while keeping the calibration scale and estimator fixed. Performance remains stable across essay, web, encyclopedic, and scientific corpora. The default Paul Graham corpus is not uniquely advantaged. RePhase therefore does not depend on a particular calibration domain. This stability supports that the offline profile captures transferable positional structure rather than corpus-specific semantics. Additional ablations on the number and length of calibration sequences are provided in Appendix A.3.

### 4.5 Matched-Budget Routing Quality

Task accuracy does not directly reveal whether an estimator selects the right KV blocks. We therefore compare routing quality under the same block budget. Retained mass measures how much

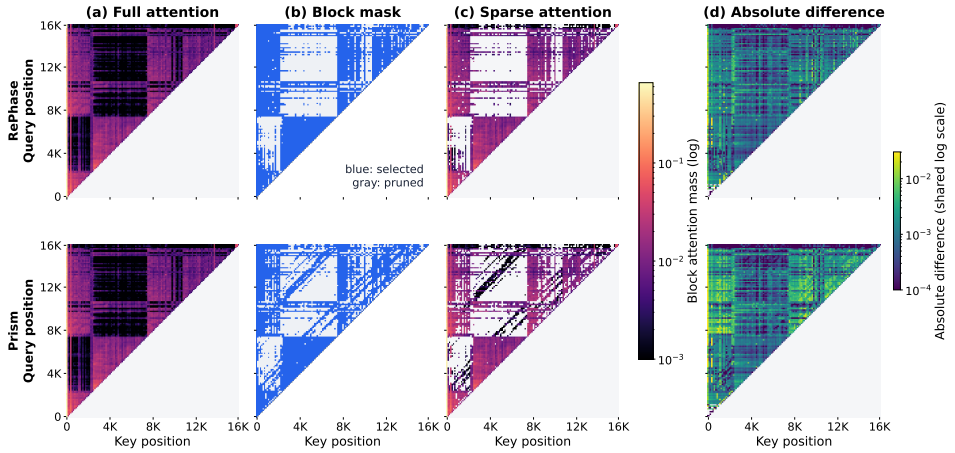



Figure 4: Attention-map comparison at 16K under the same KV-block budget. The top and bottom rows show RePhase and Prism, respectively. In this example, Prism assigns part of its budget to blocks with relatively low dense-attention mass. By contrast, the block mask selected by RePhase aligns more closely with the full-attention structure, resulting in a smaller absolute difference.

| Context | Density | Method | Mass | Recall |
|---|---|---|---|---|
| 16K | 23.61 | Prism | 80.95 | 71.16 |
| | | XAttention | 81.04 | 71.74 |
| | | **RePhase** | **81.45** | **75.92** |
| 64K | 16.64 | Prism | 75.54 | 71.62 |
| | | XAttention | 75.73 | 72.77 |
| | | **RePhase** | **76.25** | **76.58** |
| 128K | 12.37 | Prism | 71.50 | 70.41 |
| | | XAttention | 71.81 | 72.06 |
| | | **RePhase** | **72.28** | **75.58** |

Table 8: Matched-density block-routing quality. RePhase retains more dense-attention mass and achieves higher dense Top-$K$ block recall across all settings.

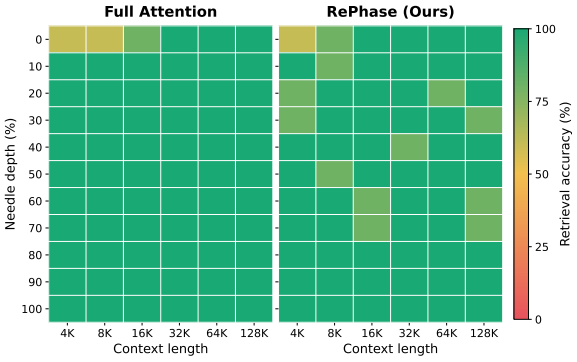



Figure 5: Needle-in-a-haystack results over input length and needle depth. RePhase stays close to full attention while using fewer attention blocks.

dense-attention mass remains after selection, while Top-$K$ recall measures the overlap with the highest-scoring dense-attention blocks. Table 8 fixes the density across methods at each context length. RePhase achieves the highest retained mass and Top-$K$ recall in every setting. Its Top-$K$ recall exceeds the strongest baseline by at least 3.52 percentage points. Because all methods select the same number of blocks, this improvement comes from better block ranking rather than a larger sparse mask. Figure 4 illustrates the same advantage, with RePhase allocating the fixed block budget more closely to the high-mass structure of full attention.

## 5 Conclusion

We introduced RePhase, a block estimator for efficient block-sparse attention. RePhase avoids the high-frequency phase cancellation induced by pooling RoPE-rotated representations by shifting content aggregation to the unrotated space and reconstructing positional structure with relative RoPE phase and a precomputed structural profile. Estimated once from forward-pass statistics without backpropagation or model-parameter updates, the profile can be reused across inference inputs for block scoring without token-level search. Experiments across long-context text and video tasks show that RePhase remains competitive with full attention while substantially reducing block-estimation overhead, providing an effective accuracy–efficiency trade-off for block-sparse attention.

## AI USE STATEMENT

Generative AI tools were used to assist with language editing and writing refinement, including improving grammar, clarity, and readability. All technical content, claims, analyses, and final wording were reviewed and verified by the authors, who take full responsibility for the content of this paper.

## REFERENCES

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, 2024.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks, 2025. URL https://arxiv.org/abs/2412.15204.

Yushi Bai, Qian Dong, Ting Jiang, Xin Lv, Zhengxiao Du, Aohan Zeng, Jie Tang, and Juanzi Li. Indexcache: Accelerating sparse attention via cross-layer index reuse. *arXiv preprint arXiv:2603.12201*, 2026.

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.

Aditya Desai, Kumar Krishna Agrawal, Shuo Yang, Alejandro Cuadron, Luis Gaspar Schroeder, Matei Zaharia, Joseph E Gonzalez, and Ion Stoica. vattention: Verified sparse attention. *arXiv preprint arXiv:2510.05688*, 2025.

Yuchen Duan, Zhe Chen, Yusong Hu, Weiyun Wang, Shenglong Ye, Botian Shi, Lewei Lu, Qibin Hou, Tong Lu, Hongsheng Li, Jifeng Dai, and Wenhai Wang. Docopilot: Improving multimodal models for document-level understanding, 2025. URL https://arxiv.org/abs/2507.14675.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 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. *arXiv preprint arXiv:2410.13276*, 2024.

Yizhao Gao, Shuming Guo, Shijie Cao, Yuqing Xia, Yu Cheng, Lei Wang, Lingxiao Ma, Yutao Sun, Tianzhu Ye, Li Dong, et al. Seerattention-r: Sparse attention adaptation for long reasoning. *arXiv preprint arXiv:2506.08889*, 2025.

Paul Graham. The age of the essay. *Paul Graham*, 2004.

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

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke

Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, David Crandall, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C. V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2024. URL https://arxiv.org/abs/2311.18259.

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.

Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding, 2024. URL https://arxiv.org/abs/2409.03420.

Junhao Hu, Fangze Li, Mingtao Xu, Feifan Meng, Shiju Zhao, Tiancheng Hu, Ting Peng, Anmin Liu, Wenrui Huang, Chenxu Liu, et al. Lil: Less is less when applying post-training sparse-attention algorithms in long-decode stage. *arXiv preprint arXiv:2601.03043*, 2026.

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.

Yinsicheng Jiang, Yeqi Huang, Liang Cheng, Cheng Deng, Xuan Sun, and Luo Mai. Contextpilot: Fast long-context inference via context reuse. *Proceedings of Machine Learning and Systems*, 8: 1639–1655, 2026.

Dongwon Jo, Beomseok Kang, Jiwon Song, and Jae-Joon Kim. Token sparse attention: Efficient long-context inference with interleaved token selection. *arXiv preprint arXiv:2602.03216*, 2026.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. *arXiv preprint arXiv:2502.20766*, 2025.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Di Liu, Ruitian Wang, Chen Chen, Mingliang Gong, Yongjie Yuan, Han Zhao, Yu Feng, Quan Chen, and Minyi Guo. Ab-sparse: Sparse attention with adaptive block size for accurate and efficient long-context inference. *arXiv preprint arXiv:2605.12110*, 2026.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4969–4983, 2020.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, et al. Moba: Mixture of block attention for long-context llms. *Advances in Neural Information Processing Systems*, 38:17790–17815, 2026.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12585–12602, 2024.

Weian Mao, Xi Lin, Wei Huang, Yuxin Xie, Tianfu Fu, Bohan Zhuang, Song Han, and Yukang Chen. Triattention: Efficient long reasoning with trigonometric kv compression, 2026. URL https://arxiv.org/abs/2604.04921.

Pierre-Emmanuel Mazaré, Gergely Szilvasy, Maria Lomeli, Francisco Massa, Naila Murray, Hervé Jégou, and Matthijs Douze. Inference-time sparse attention with asymmetric indexing. *arXiv preprint arXiv:2502.08246*, 2025.

Piotr Nawrot, Robert Li, Renjie Huang, Sebastian Ruder, Kelly Marchisio, and Edoardo M Ponti. The sparse frontier: Sparse attention trade-offs in transformer llms. *arXiv preprint arXiv:2504.17768*, 2025.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 31932–31951, 2024.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

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

Ziyao Tang, Pengkun Jiao, Xinhang Chen, Wei Liu, Shiyong Li, and Jingjing Chen. Predicting future utility: Global combinatorial optimization for task-agnostic kv cache eviction, 2026. URL https://arxiv.org/abs/2602.08585.

V Team, Wenyi Hong, Xiaotao Gu, Ziyang Pan, Zhen Yang, Yuting Wang, Yue Wang, Yuanchang Yue, Yu Wang, Yanling Wang, Yan Wang, Xijun Liu, Wenmeng Yu, Weihan Wang, Wei Li, Shuaiqi Duan, Sheng Yang, Ruiliang Lv, Mingdao Liu, Lihang Pan, Ke Ning, Junhui Ji, Jinjiang Wang, Jing Chen, Jiazheng Xu, Jiale Zhu, Jiale Cheng, Ji Qi, Guobing Gan, Guo Wang, Cong Yao, Zijun Dou, Zihao Zhou, Zihan Wang, Zhiqi Ge, Zhijie Li, Zhenyu Hou, Zhao Xue, Zehui Wang, Zehan Qi, Zehai He, Yutao Zhang, Yusen Liu, Yukuo Cen, Yuchen Li, Yuan Wang, Yu Yang, Yongbin Liu, Yijian Lu, Yifan Xu, Yanzi Wang, Yanxiao Zhao, Yanfeng Wang, Yadong Xue, Yabo Xu, Xinyu Zhang, Xinyu Liu, Xiao Liu, Wenyi Zhao, Wenkai Li, Tianyu Tong, Tianshu Zhang, Shudan Zhang, Shengdong Yan, Qinkai Zheng, Mingde Xu, Licheng Bao, lat Long long, Jiaxing Xu, Jiaxin Fan, Jiawen Qian, Jiali Chen, Jiahui Lin, Jiadai Sun, Haozhi Zheng, Haoran Wang, Haochen Li, Hanyu Lai, Han Xu, Fan Yang, Dan Zhang, Da Yin, Chuangxin Zhao, Chengcheng Wu, Boyan Shi, Bowen Lv, Bowei Jia, Bo Li, Bin Chen, Baoxu Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-5v-turbo: Toward a native foundation model for multimodal agents, 2026. URL https://arxiv.org/abs/2604.26752.

Xinghao Wang, Pengyu Wang, Dong Zhang, Chenkun Tan, Shaojun Zhou, Zhaoxiang Liu, Shiguo Lian, Fangxu Liu, Kai Song, and Xipeng Qiu. Sparser block-sparse attention via token permutation. *arXiv preprint arXiv:2510.21270*, 2025.

Xinghao Wang, Pengyu Wang, Xiaoran Liu, Fangxu Liu, Jason Chu, Kai Song, and Xipeng Qiu. Prism: Spectral-aware block-sparse attention. *arXiv preprint arXiv:2602.08426*, 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Jianzong Wu, Liang Hou, Haotian Yang, Xin Tao, Ye Tian, Pengfei Wan, Di Zhang, and Yunhai Tong. Vmoba: Mixture-of-block attention for video diffusion models. *arXiv preprint arXiv:2506.23858*, 2025.

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, 2024.

Di Xiu, Hongyin Tang, Bolin Rong, Lizhi Yan, Jingang Wang, Yifan Lu, and Xunliang Cai. A preliminary study on the promises and challenges of native top-$k$ sparse attention. *arXiv preprint arXiv:2512.03494*, 2025.

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

Yufei Xu, Fanxu Meng, Fan Jiang, Yuxuan Wang, Ruijie Zhou, Zhaohui Wang, Jiexi Wu, Zhixin Pan, Xiaojuan Tang, Wenjie Pei, et al. Hisa: Efficient hierarchical indexing for fine-grained sparse attention. *arXiv preprint arXiv:2603.28458*, 2026.

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

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, 2025.

Zhiyuan Zeng, Qipeng Guo, Xiaoran Liu, Zhangyue Yin, Wentao Shu, Mianqiu Huang, Bo Wang, Yunhua Zhou, Linlin Li, Qun Liu, et al. Memorize step by step: Efficient long-context prefilling with incremental memory and decremental chunk. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21021–21034, 2024.

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

Ruijie Zhou, Fanxu Meng, Yufei Xu, Tongxuan Liu, Guangming Lu, Muhan Zhang, and Wenjie Pei. Misa: Mixture of indexer sparse attention for long-context llm inference. *arXiv preprint arXiv:2605.07363*, 2026.

# A Additional Experimental Details and Results

## A.1 Implementation and Evaluation Setup

To ensure direct comparability, we strictly follow the evaluation protocol of Prism Wang et al. (2026) for all overlapping experiments. We use 128-token blocks and deterministic top-$p$ block selection with $p = 0.9$ in all main experiments unless stated otherwise. We apply the causal and padding masks before score normalization and block selection. One sink block and the most recent local block participate in the top-$p$ selection. If either block is absent from the selected set, we add it as a fallback and include it in the reported density. The selected blocks are processed by an exact block-sparse attention kernel. We implement both block estimation and sparse attention using custom Triton kernels.

We use FlashAttention Dao et al. (2022) as the exact-attention reference. We compare RePhase with five representative sparse-attention methods: StreamingLLM Xiao et al. (2024), MInference Jiang et al. (2024), FlexPrefill Lai et al. (2025), XAttention Xu et al. (2025), and Prism Wang et al. (2026). We use the official implementations and recommended configurations whenever available.

For Qwen3-8B and Qwen3-30B-A3B, we apply $4\times$ YaRN Peng et al. (2024). Llama-3.1-8B-Instruct Grattafiori et al. (2024) and Qwen3.8-27B are evaluated within their native context windows. For Qwen3-30B-A3B Yang et al. (2025), RePhase modifies only block estimation in the attention layers and leaves expert routing unchanged. For Qwen3.8-27B, we apply RePhase only to its gated-attention layers and leave the Gated DeltaNet layers unchanged. Within each gated-attention head, positional reconstruction is applied to the rotary subspace, while the non-rotary dimensions retain their original contribution to the block score. For Qwen3-VL-8B-Instruct, we preserve the native interleaved M-RoPE position indices.

We use the official Hugging Face checkpoints and their corresponding tokenizers. All evaluations use a batch size of one and BF16 precision. We follow the official prompts, preprocessing pipelines, and evaluation metrics for all benchmarks. We use the official baseline implementations and retain their recommended settings whenever available. For controlled block-budget comparisons, all methods use the same selection budget.

We strictly follow the efficiency-measurement protocol of Prism Wang et al. (2026). Efficiency measurements are conducted on a single NVIDIA H100 80GB GPU. Following the Prism protocol Wang et al. (2026), we perform three warm-up runs followed by five measured runs. We use CUDA events with device synchronization and report the mean latency. Reported attention-prefill latency includes block estimation and sparse attention execution but excludes the one-time offline calibration procedure. Our software environment uses PyTorch 2.10.0+cu129, Triton 3.6.0, and PyTorch CUDA 12.9.

## A.2 Empirical Observation of High-Frequency Phase Cancellation

We provide direct empirical evidence for the phase cancellation discussed in the Introduction. Using actual activations from Llama-3.1-8B-Instruct, we measure the root-mean-square (RMS) energy of each rotary frequency subspace before and after block pooling. This experiment uses model representations rather than a constructed signal and reveals a consistent trend across the sampled data.

Figure 6 shows that RoPE-rotated pooling strongly attenuates the high-frequency subspaces. Their phases change rapidly across token positions. Averaging these rotated vectors therefore causes destructive interference and removes much of their energy. The loss becomes smaller toward lower frequencies because their phases vary more slowly within a block. In contrast, unrotated pooling avoids mixing position-dependent phases and closely preserves the energy of the original token representations. This observation motivates RePhase to aggregate content in the unrotated space and reconstruct positional structure afterward.

## A.3 Offline Profile Construction

We construct a separate positional profile for each model checkpoint. For the text models, we sample 1,000 unlabeled sequences of length 16K from the Paul Graham Essays corpus. The calibration data

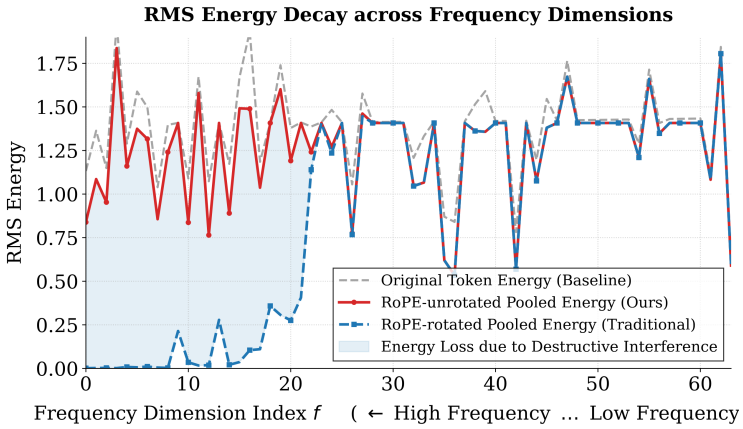



Figure 6: Empirical RMS energy across rotary frequency dimensions in Llama-3.1-8B-Instruct. RoPE-rotated pooling strongly attenuates high-frequency components, whereas unrotated pooling closely preserves the energy of the original token representations. The shaded region denotes the energy lost through destructive phase interference.

require no task labels or benchmark examples. Each profile uses the same positional-encoding configuration and block size as inference.

We run the model on the calibration sequences and collect the unrotated interaction $X_{s,\ell,h,f}(i,j)$ and the demodulated phase-aware interaction $Y_{s,\ell,h,f}(i,j)$. For each layer, head, rotary frequency, and relative block displacement, we accumulate the sufficient statistics $\sum YX^*$ and $\sum |X|^2$. We then compute the regularized complex coefficient in Equation 3. This procedure uses only forward passes. It requires neither backpropagation nor model-parameter updates.

Profile construction is performed once and requires only forward-pass statistics. Using 1,000 sequences of length 16K, it takes approximately 20–25 minutes on a single NVIDIA H100 80GB GPU. For $N_\ell$ layers, $H$ attention heads, $d_{\mathrm{R}}$ rotary dimensions, and $D_\Delta$ calibrated displacements, the profile contains $N_\ell H d_{\mathrm{R}} D_\Delta$ real-valued coefficients. Under our default configuration, the Qwen3-8B profile contains 18.9M coefficients and occupies 36 MiB in BF16. The profile is fixed after construction and reused across all inference inputs, making both its one-time calibration cost and storage overhead small.

For Qwen3-VL-8B-Instruct, we construct a separate profile using videos from VideoChatGPT Maaz et al. (2024). We process these videos with the native Qwen3-VL preprocessing pipeline and collect the same statistics from the resulting interleaved visual–text sequences. We preserve the native interleaved M-RoPE position indices when constructing the profile. The same position indices are used to evaluate the analytical relative phase during inference. We do not use examples from VideoMME or LongVideoBench for calibration.

The resulting profile is fixed during evaluation and reused across all inference sequences. For distances covered by calibration, RePhase directly indexes the corresponding coefficient. For positions beyond the calibration range, we clamp only the profile index to the largest calibrated displacement. The analytical RoPE phase is still evaluated at the actual position. This preserves the relative positional geometry while extrapolating only the profiled within-block response.

The profile depends on the model checkpoint, positional-encoding configuration, and block size, but not on the downstream task. We test this property using four text corpora with distinct content distributions. Paul Graham Essays is our default corpus and contains curated long-form essays Graham (2004). FineWeb provides broad web text collected and filtered from Common Crawl Penedo et al. (2024). Wikipedia represents encyclopedic prose, while peS2o contains open-access scientific papers derived from S2ORC Lo et al. (2020). Together, these sources cover essay, web, encyclopedic, and scientific text.

For each corpus, we keep the number and length of calibration sequences fixed. We also use the same model checkpoint, block size, estimator, and profile-fitting procedure. The calibration corpus

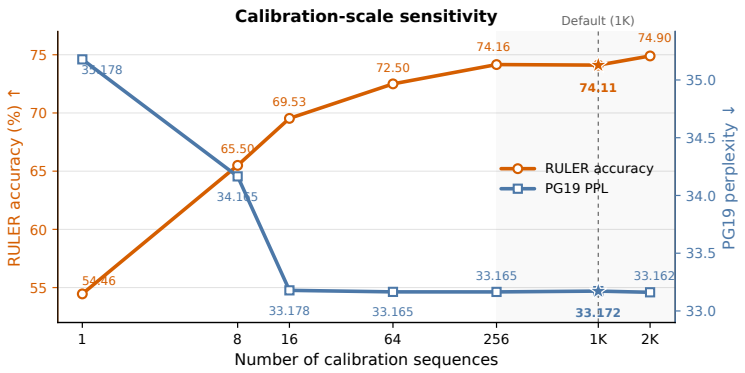



Figure 7: Sensitivity to the number of calibration sequences on Qwen3-8B at 128K. All sequences contain 16K tokens and are sampled from the Paul Graham Essays corpus. Performance stabilizes after the first few calibration sequences. The dashed line marks the default setting of 1,000 sequences.

is therefore the only changed variable. Table 7 reports the resulting RULER accuracy and PG19 perplexity. The stable performance across these sources shows that the positional profile is not tied to corpus-specific semantics.

### A.4 Evaluation Metrics

We follow the official metrics of each benchmark. RULER Hsieh et al. (2024) reports accuracy at each context length. LongBench Bai et al. (2024; 2025) reports the official task scores and their category averages. PG19 Rae et al. (2019) reports token-level perplexity. VideoMME Fu et al. (2025) and LongVideoBench Wu et al. (2024) report multiple-choice accuracy.

We additionally measure block-routing quality against dense attention. Let $A_{s,\ell,h,t,u}$ denote the dense-attention probability from query token $t$ to key token $u$ for sequence $s$, layer $\ell$, and head $h$, after applying the causal and padding masks. For query block $\mathcal{Q}_i$ and key block $\mathcal{K}_j$, we define their dense-attention mass as

$$m_{s,\ell,h}(i,j) = \frac{1}{|\mathcal{Q}_i|} \sum_{t \in \mathcal{Q}_i} \sum_{u \in \mathcal{K}_j} A_{s,\ell,h,t,u}. \tag{5}$$

Thus, $m_{s,\ell,h}(i,j)$ measures the average dense-attention probability assigned to key block $j$ by query block $i$.

Let $\mathcal{S}_{s,\ell,h,i}$ be the set of key blocks selected by a routing method. Its retained attention mass is

$$\text{Mass} = \mathbb{E}_{s,\ell,h,i} \left[ \sum_{j \in \mathcal{S}_{s,\ell,h,i}} m_{s,\ell,h}(i,j) \right]. \tag{6}$$

A larger value means that the selected blocks preserve more of the probability mass assigned by dense attention.

For each query block, we rank all valid key blocks by $m_{s,\ell,h}(i,j)$. Let $\mathcal{T}_K(s,\ell,h,i)$ contain the $K$ blocks with the largest dense-attention mass, where $K = |\mathcal{S}_{s,\ell,h,i}|$. Dense Top-$K$ block recall is

$$\text{Recall@}K = \mathbb{E}_{s,\ell,h,i} \left[ \frac{|\mathcal{S}_{s,\ell,h,i} \cap \mathcal{T}_K(s,\ell,h,i)|}{K} \right]. \tag{7}$$

It measures how often the router selects the blocks preferred by dense attention. In matched-budget comparisons, all methods use the same $K$ for each query block. Their recall differences therefore reflect routing quality rather than different selection budgets.

We compute realized density as the number of selected valid causal block pairs divided by the total number of valid causal block pairs. Padded block pairs are excluded from both counts. Any sink or local block added by the fallback rule is included in the selected set and therefore contributes to the realized density.

| Block Size | RULER Acc. (%) ↑ | PG19 PPL ↓ |
|---|---|---|
| 64 | **75.27** | 33.199 |
| **128 (default)** | 74.11 | **33.172** |
| 256 | 62.73 | 33.180 |

Table 9: Sensitivity to block size on Qwen3-8B at 128K. Smaller blocks provide finer routing granularity, while the default block size of 128 achieves the lowest PG19 perplexity.

| Calibration Length | RULER Acc. (%) ↑ | PG19 PPL ↓ |
|---|---|---|
| 8K | 74.267 | **33.164** |
| **16K (default)** | 74.110 | 33.172 |
| 32K | **74.463** | 33.170 |
| 64K | 74.099 | 33.172 |

Table 10: Sensitivity to calibration sequence length on Qwen3-8B at 128K. All settings use the same number of calibration sequences. Both RULER accuracy and PG19 perplexity remain stable across calibration lengths.

### A.5 Additional Ablation Results

Figure 7 studies the effect of calibration-set size. Increasing the number of sequences from 1 to 16 raises RULER accuracy from 54.46 to 69.53 and reduces PG19 perplexity from 35.178 to 33.178. Beyond 64 sequences, PG19 perplexity remains nearly unchanged, whereas RULER accuracy continues to improve with diminishing returns. RULER accuracy increases from 72.50 with 64 sequences to 74.11 with 1,000 sequences and 74.90 with 2,000 sequences, while PG19 perplexity changes by at most 0.010 over this range. We therefore use 1,000 sequences as a practical trade-off between calibration cost and accuracy. Overall, the results show diminishing returns as the calibration set grows.

We next vary the calibration sequence length from 8K to 64K while keeping the number of calibration sequences fixed. As shown in Table 10, RULER accuracy varies by only 0.36 points across all four settings. PG19 perplexity is even more stable, with a maximum difference of 0.008. The best values of the two metrics occur at different calibration lengths. RePhase therefore does not rely on a particular sequence length for offline profiling. We use 16K as the default because it covers a broad range of relative block displacements without unnecessarily increasing calibration cost.

We finally vary the block size while keeping all other settings fixed. Table 9 shows that 64-token blocks achieve the highest RULER accuracy, exceeding the default setting by 1.16 points. PG19 perplexity remains stable across the three settings, with 128-token blocks obtaining the lowest value. The effect of block size is therefore more pronounced on retrieval-sensitive routing than on aggregate language-modeling fidelity. Increasing the block size to 256 substantially reduces RULER accuracy. We therefore use 128-token blocks as a balance between routing granularity, language-modeling fidelity, and execution efficiency. This trend is consistent with the empirical observation in Figure 1(c). The positional-response prediction error is largest when the rotary wavelength is shorter than the block size. Increasing the block size moves more rotary subspaces into this high-error regime. It also makes each routing decision coarser. Both effects make fine-grained positional structure harder to preserve.

### A.6 Additional Results on Larger Models

We further evaluate RePhase on Qwen3-30B-A3B with YaRN and Qwen3.8-27B. The former tests a larger mixture-of-experts architecture. RePhase changes only its attention estimator and leaves expert routing unchanged. The latter contains both gated-attention and Gated DeltaNet layers. We apply RePhase only to the gated-attention layers. These settings test whether block-level positional reconstruction remains effective when it controls only part of a larger inference pipeline.

Table 11 reports the LongBench results. On Qwen3-30B-A3B, RePhase reaches 40.5, achieving the best sparse result and remaining within 0.6 points of full attention. It obtains the best sparse

| | *Qwen3-30B-A3B with YaRN* | | | | | | | *Qwen3.8-27B* | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **S-Doc** | **M-Doc** | **Summ.** | **Few** | **Code** | **Synth.** | **Avg.** | **S-Doc** | **M-Doc** | **Summ.** | **Few** | **Code** | **Synth.** | **Avg.** |
| FlashAttention | 48.1 | 40.4 | 24.2 | 62.8 | 2.9 | 68.2 | 41.1 | 51.4 | 54.9 | 23.5 | 66.8 | 57.5 | 72.1 | 54.4 |
| StreamingLLM | 33.5 | 26.3 | 22.7 | 42.0 | 7.3 | 40.9 | 28.8 | 31.3 | 30.8 | 20.5 | 51.9 | 36.5 | 26.7 | 32.9 |
| MInference | 44.7 | 36.4 | 23.6 | 60.6 | 3.1 | 49.7 | 36.4 | 45.0 | 48.7 | 22.9 | 65.1 | 49.0 | 59.5 | 48.4 |
| FlexPrefill | 45.7 | 38.8 | 24.0 | 61.0 | **4.0** | 51.7 | 37.5 | **52.0** | 54.8 | 23.4 | **66.7** | 56.0 | 66.4 | 53.2 |
| XAttention | **47.2** | 38.8 | **24.1** | 62.0 | 3.6 | 66.6 | 40.4 | 51.6 | 54.5 | **23.5** | **66.7** | **59.1** | **71.6** | **54.5** |
| Prism | 46.9 | **39.7** | 24.0 | 62.7 | 2.7 | 66.3 | 40.4 | 51.4 | 54.5 | 23.4 | 66.4 | 58.2 | 71.4 | 54.2 |
| **RePhase (Ours)** | 46.9 | 39.4 | **24.1** | **62.9** | 2.7 | **67.0** | **40.5** | 51.3 | **54.9** | **23.5** | 66.5 | 58.4 | **71.6** | 54.4 |

Table 11: LongBench scores on larger and hybrid models. S-Doc, M-Doc, Few, and Synth. denote single-document QA, multi-document QA, few-shot learning, and synthetic tasks. Bold and underlined values denote the best and second-best results among sparse-attention methods, respectively.

scores on few-shot learning and synthetic tasks, and ties the best result on summarization. On Qwen3.8-27B, RePhase reaches 54.4, matching full attention and remaining within 0.1 points of the strongest sparse method. It achieves the best sparse result on multi-document QA, ties the best result on summarization, and exceeds full attention on code. No method dominates every category. RePhase nevertheless remains strong across tasks that require different attention patterns, rather than specializing in a single type of long-context retrieval.

LongBench-v2 further separates task difficulty from context length (Table 12). On Qwen3-30B-A3B, RePhase reaches 32.0, matching full attention and achieving the best sparse result. It leads or ties the sparse methods on the Easy, Hard, Short, and Medium subsets. It also exceeds full attention by 1.9 points on the Long subset. On Qwen3.8-27B, RePhase reaches 51.9, outperforming full attention by 1.2 points and ranking second among sparse methods. Its gains over full attention appear on the Easy, Hard, Short, and Long subsets, with the largest gain of 2.8 points on Long examples. These results extend the main-paper findings across model scale and architecture. They also suggest that selective block computation can remove distracting context instead of merely approximating dense attention.

### A.7 Qualitative Analysis of Block Routing

Figure 9 visualizes the actual block masks selected by RePhase on four 16K inputs. Each dense-attention map contains a strong local diagonal together with input-dependent long-range structures. The selected masks follow both components. They preserve the local region without collapsing into a fixed sliding window, while retaining the sparse off-diagonal bands that carry long-range attention.

The masks also change substantially across inputs. This variation shows that RePhase performs content-dependent routing rather than applying a fixed sparse pattern. The resulting sparse-attention maps closely reproduce the dominant structures of full attention. Their residual differences are concentrated in regions with low dense attention mass.

Across the four examples, RePhase prunes 71.8–74.8% of the causal block pairs while retaining 99.57–99.70% of the dense-attention mass. The attention-map cosine similarity remains above 0.99998, and the attention-output cosine similarity remains above 0.9966. These results connect block-level routing quality to output fidelity: RePhase removes mostly negligible interactions while preserving the blocks that determine the attention output.

Table 12 shows the same trend on LongBench-v2. RePhase obtains the best sparse result of 32.0 on Qwen3-30B-A3B, matching full attention. On Qwen3.8-27B, it reaches 51.9. It exceeds full attention by 1.2 points and ranks second among sparse methods. The consistent results across both benchmarks show that RePhase remains effective beyond the 8B models used in the main evaluation.

### A.8 Efficiency–Performance Pareto Frontier

The main results evaluate accuracy and latency separately. Figure 8 combines both dimensions at 128K. RePhase lies above and to the left of every sparse baseline. It therefore achieves both higher RULER accuracy and lower attention-prefill latency. MInference approaches RePhase in latency but loses more accuracy. Prism and FlexPrefill retain stronger accuracy than the remaining baselines,

| Method | *Qwen3-30B-A3B with YaRN* | | | | | | *Qwen3.8-27B* | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | E | H | S<32K | M<128K | L>128K | Overall | E | H | S<32K | M<128K | L>128K | Overall |
| FlashAttention | 34.4 | 30.5 | 35.6 | 32.1 | 25.9 | 32.0 | 53.6 | 48.9 | 53.9 | 51.6 | 43.5 | 50.7 |
| StreamingLLM | 31.2 | **30.9** | 33.9 | 26.0 | **36.1** | 31.0 | 33.9 | 34.1 | 35.0 | 32.6 | 35.2 | 34.0 |
| MInference | 31.2 | 29.3 | 35.0 | 26.0 | 29.6 | 30.0 | 45.8 | 39.9 | 42.2 | 40.5 | 45.4 | 42.1 |
| FlexPrefill | 33.3 | 30.5 | 33.9 | 30.2 | 30.6 | 31.6 | 53.6 | 48.6 | 53.3 | 49.3 | **48.1** | 50.5 |
| XAttention | 29.7 | 28.3 | 30.0 | 28.4 | 27.8 | 28.8 | **57.8** | **49.2** | 55.0 | **53.5** | 46.3 | **52.5** |
| Prism | **33.9** | 26.7 | 33.3 | 27.9 | 25.9 | 29.4 | 54.2 | 48.9 | 55.6 | 51.2 | 42.6 | 50.9 |
| **RePhase (Ours)** | **33.9** | **30.9** | **36.1** | **30.7** | 27.8 | **32.0** | 56.2 | **49.2** | **56.1** | 51.2 | 46.3 | 51.9 |

Table 12: LongBench-v2 accuracy on larger and hybrid models. E and H denote Easy and Hard, while S, M, and L denote Short, Medium, and Long contexts. Context lengths are measured in words. Bold and underlined values denote the best and second-best results among sparse-attention methods, respectively.

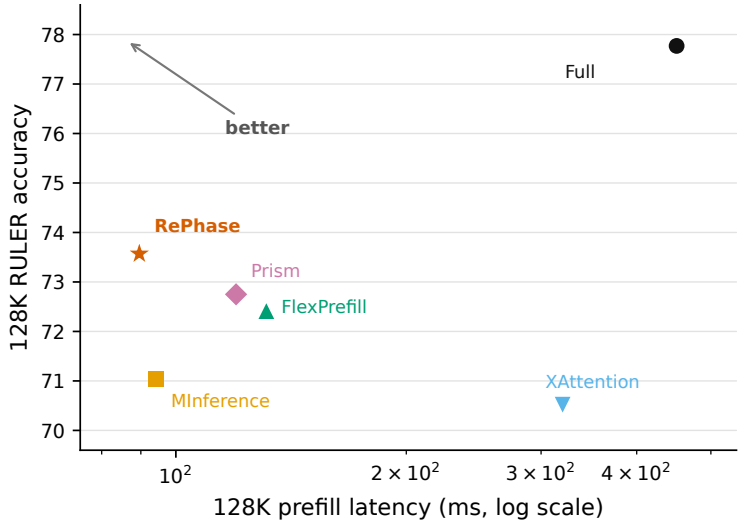



Figure 8: Efficiency–performance trade-off at 128K. RULER accuracy is plotted against end-to-end attention-prefill latency on a single NVIDIA H100 80GB GPU. Latency includes block estimation and sparse attention execution. The upper-left direction is better. RePhase dominates all sparse baselines and achieves a 5.03× speedup over full attention.

but require higher latency. XAttention incurs the largest sparse-attention latency at this length. Full attention remains more accurate, but requires substantially more computation. RePhase reaches a 5.03× speedup over full attention while preserving much of its retrieval accuracy. These results show that lightweight block estimation improves the accuracy–efficiency Pareto frontier rather than exchanging routing quality for lower latency.

# B Derivations for RePhase

## B.1 Exact Relative-Phase Factorization

In complex form, RoPE maps the query and key at position $r$ as $q^{\mathrm{R}}_{r,f} = q_{r,f}e^{\mathrm{i}\omega_f r}$ and $k^{\mathrm{R}}_{r,f} = k_{r,f}e^{\mathrm{i}\omega_f r}$. For token positions $iB+n$ and $jB+m$, their interaction in rotary subspace $f$ is

$$q^{\mathrm{R}}_{iB+n,f}\left(k^{\mathrm{R}}_{jB+m,f}\right)^* = q_{iB+n,f}\left(k_{jB+m,f}\right)^* e^{\mathrm{i}\omega_f B\Delta}e^{\mathrm{i}\omega_f(n-m)}. \tag{8}$$

The factor $e^{\mathrm{i}\omega_f B\Delta}$ depends only on the relative block displacement, while $e^{\mathrm{i}\omega_f(n-m)}$ depends on the within-block offsets. Averaging Equation 8 over $n$ and $m$, then summing the real contribution over rotary subspaces, gives Equation 1. The factorization is exact.

### B.2 Closed Form of the Positional Response

Fix $\ell$, $h$, $f$, and $\Delta$. For brevity, write $X_{sij} = X_{s,\ell,h,f}(i,j)$ and $Y_{sij} = Y_{s,\ell,h,f}(i,j)$, and let $M = |\mathcal{P}_{\mathcal{D},\Delta}|$. The empirical positional response minimizes

$$J(c) = \frac{1}{M} \sum_{(s,i,j)\in\mathcal{P}_{\mathcal{D},\Delta}} |Y_{sij} - cX_{sij}|^2 + \lambda|c|^2. \tag{9}$$

Using Wirtinger differentiation with respect to $c^*$ gives

$$\frac{\partial J}{\partial c^*} = -\frac{1}{M} \sum_{(s,i,j)\in\mathcal{P}_{\mathcal{D},\Delta}} X^*_{sij} \left(Y_{sij} - cX_{sij}\right) + \lambda c. \tag{10}$$

Setting this derivative to zero yields

$$c \left( \sum_{(s,i,j)\in\mathcal{P}_{\mathcal{D},\Delta}} |X_{sij}|^2 + M\lambda \right) = \sum_{(s,i,j)\in\mathcal{P}_{\mathcal{D},\Delta}} Y_{sij} X^*_{sij}. \tag{11}$$

Solving Equation 11 for $c$ and restoring the full indices gives Equation 3.

### B.3 Equivalent Real-Valued Block Score

We expand Equation 4 in polar form. Write

$$\begin{aligned} \hat{\mathcal{C}}_f(\Delta) &= \hat{\Lambda}_f(\Delta) e^{\mathrm{i}\hat{\theta}_f(\Delta)}, \\ \bar{q}_{i,f} &= \rho^q_{i,f} e^{\mathrm{i}\alpha^q_{i,f}}, \\ \bar{k}_{j,f} &= \rho^k_{j,f} e^{\mathrm{i}\alpha^k_{j,f}}, \end{aligned} \tag{12}$$

where $\rho^q_{i,f} = |\bar{q}_{i,f}|$, $\rho^k_{j,f} = |\bar{k}_{j,f}|$, and $\hat{\Lambda}_f(\Delta) = |\hat{\mathcal{C}}_f(\Delta)|$. Equation 2 gives $X_f(i,j) = \rho^q_{i,f}\rho^k_{j,f} e^{\mathrm{i}(\alpha^q_{i,f} - \alpha^k_{j,f})}$. Substituting these terms into Equation 4 and using $\mathrm{Re}(e^{\mathrm{i}\psi}) = \cos(\psi)$ gives

$$\begin{aligned} S_{\mathrm{RePhase}}(i,j) = \frac{1}{\sqrt{d}} \sum_{f=1}^{d/2} &\hat{\Lambda}_f(\Delta) \rho^q_{i,f} \rho^k_{j,f} \\ &\cdot \cos\left(\omega_f B\Delta + \hat{\theta}_f(\Delta) + \alpha^q_{i,f} - \alpha^k_{j,f}\right). \end{aligned} \tag{13}$$

Equation 13 is algebraically equivalent to Equation 4. The amplitude $\hat{\Lambda}_f(\Delta)$ and phase $\hat{\theta}_f(\Delta)$ correspond to the two profile components used in the component ablation.

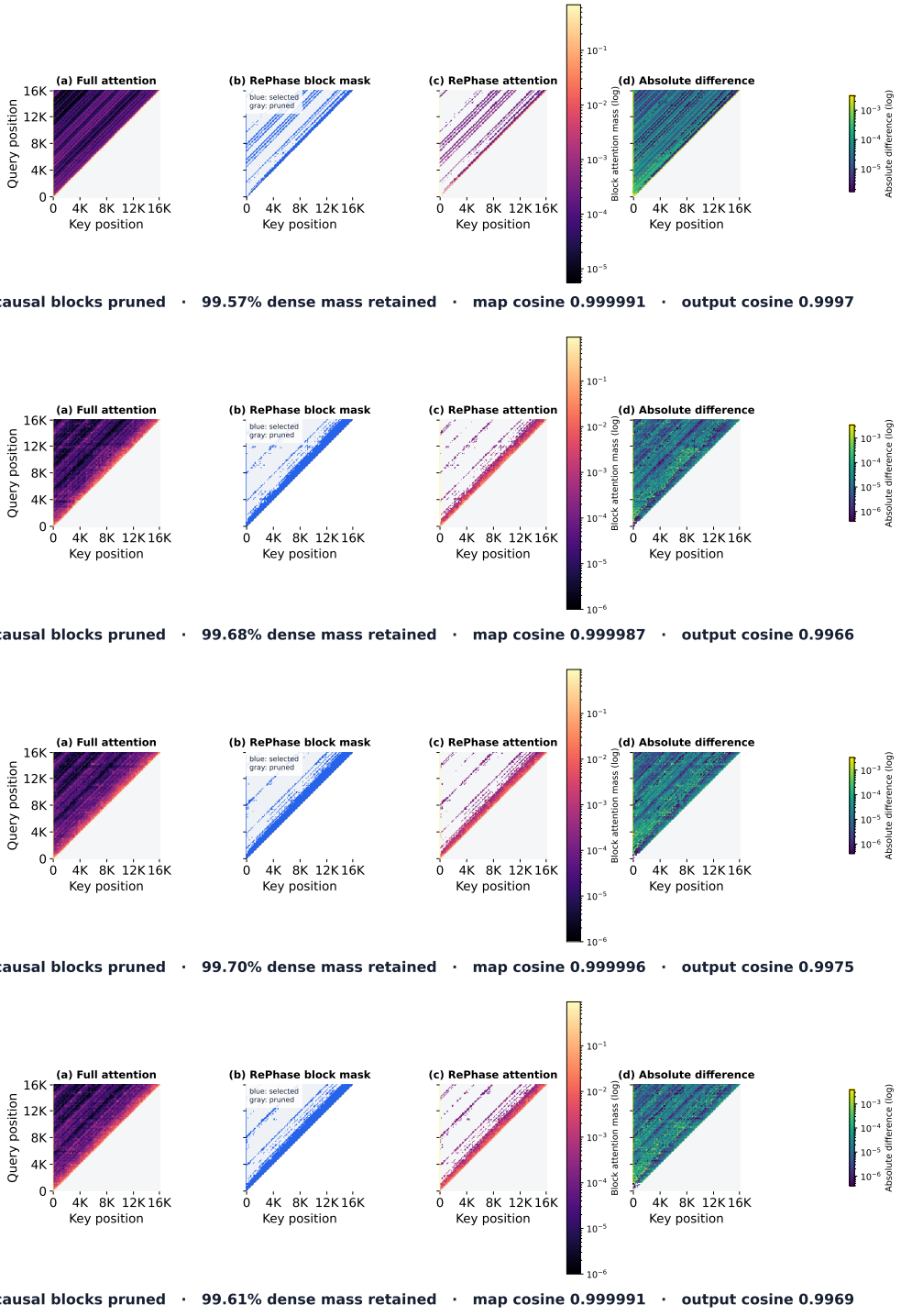



Figure 9: RePhase routing on four representative 16K inputs. The selected masks adapt to different dense-attention structures while preserving both local diagonals and input-dependent long-range bands. RePhase prunes 71.8–74.8% of the causal block pairs but retains 99.57–99.70% of the dense-attention mass. The resulting attention maps achieve cosine similarity above 0.99998, while the attention outputs remain above 0.9966. The residual differences are concentrated in low-mass regions, showing that RePhase removes mostly negligible interactions rather than uniformly approximating the dense map.