ECHOSTRIPE ATTENTION: CROSS-LAYER STRIPE MEMORY FOR EFFICIENT LONG-CONTEXT INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-context inference is constrained by the quadratic cost of self-attention, whereas aggressive sparsification can remove structured long-range interactions needed by downstream tasks. We focus on lag-indexed diagonal stripes in the attention matrix: query–key pairs with the same relative offset share a common lag and can be represented compactly by a one-dimensional profile. Existing dynamic routing typically treats each layer as an independent decision, making the resulting sparse route sensitive to local fluctuations. We introduce *EchoStripe*, a training-free framework for stateful cross-layer sparse routing. Its *Stripe Memory* recursively propagates a compact preceding-layer lag-profile state and fuses it with current-layer evidence using a head-specific reliability-aware weight. EchoStripe further applies *Mask Refinement*, a bounded token-guided procedure that corrects the complete causal base mask through conservative additions and pruning. Across LongBench with Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct, and InfiniteBench with Llama-3.1-8B-Instruct, EchoStripe remains within 0.42, 0.85, and 1.16 points of the strongest sparse baseline, respectively. At 128K tokens, it reduces measured attention-operator latency by $5.20\times$ relative to MInference and achieves a $7.73\times$ speedup over Full Attention. On the common 20-dataset LongBench suite, Stripe Memory and Mask Refinement jointly improve the current-layer-only variant by 5.90 points. These results show that cross-layer routing state provides a favorable accuracy–efficiency operating point for long-context inference.

1 INTRODUCTION

Dense self-attention scales quadratically with sequence length (Vaswani et al., 2017), making long documents, code repositories, and multi-document inputs expensive (Bai et al., 2024; Tay et al., 2021; Hsieh et al., 2024). Exact IO-aware kernels improve utilization but still evaluate all token pairs (Dao et al., 2022; Dao, 2024). Sparse prefill methods instead retain structured, input-dependent interactions (Jiang et al., 2024; Lai et al., 2025; Xu et al., 2025).

We focus on diagonal stripes in the pre-softmax attention matrix $QK^{\top}/\sqrt{d_h}$. Query–key pairs (i, j) sharing a relative offset $d = i - j$ lie on the same diagonal and can therefore be represented by a single lag index. Frequency-domain lag routing has shown that this structure can supply efficient current-layer sparse-routing evidence (Anonymous, 2026). However, most dynamic selectors still make a new, layer-isolated discrete decision from the current layer alone. Selected lags often recur across neighboring layers: a route that is temporarily weak at one layer may have strong preceding-layer support, whereas a transient current-layer route may not persist.

We therefore formulate sparse routing as a *depth-aware causal state-estimation problem*. EchoStripe first classifies whether a head exhibits stripe-like block-level directional structure. For stripe heads, it extracts a current lag profile and combines it with a recursively propagated *Stripe Memory* profile. The fused profile selects the same number of lags as the current-layer selector, so memory changes routing quality rather than enlarging the stripe-route budget. EchoStripe then forms a complete causal base mask from the selected route, a complementary column route, and deterministic sink, recent, and local connections. Finally, *Mask Refinement* uses bounded current-layer token–block evidence to conservatively add promising unselected blocks and prune weak dynamic blocks.

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.

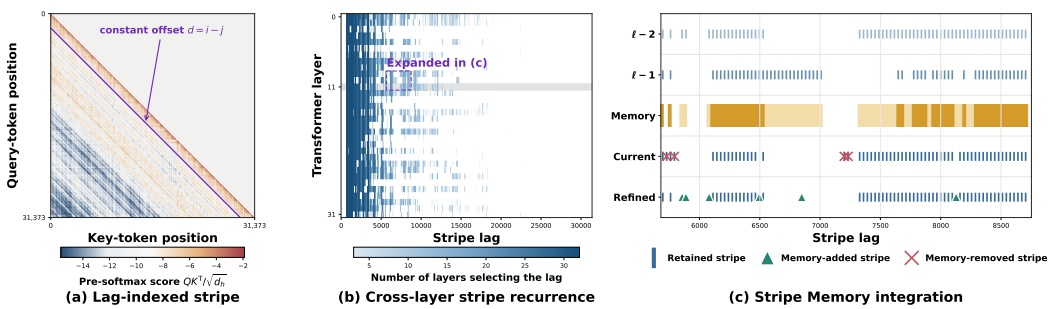


Figure 1: **From lag structure to causal cross-layer routing.** **(a)** Equal-lag token pairs form a diagonal stripe. **(b)** Selected lag routes recur across Transformer layers. **(c)** Stripe Memory integrates preceding evidence with the current-layer decision: retained routes are supported by both signals, while green triangles and red crosses denote memory-added and memory-removed routes, respectively.

Contributions and related work. Our contributions are threefold. First, we formulate lag-structured sparse attention as a depth-aware causal routing problem and maintain preceding routing evidence as a compact recursive one-dimensional state, rather than caching token-level attention maps or discarding the route after every layer. Second, we introduce reliability-aware Stripe Memory, whose head-specific adaptive fusion depends on historical stability, current-route confidence, and directional structure while preserving the current stripe-route budget. Third, we introduce bounded Mask Refinement, which turns route-level evidence into conservative token-guided corrections of the complete causal mask. We evaluate EchoStripe at matched realized keep ratios on LongBench and InfiniteBench, and verify that its stateful routing rule transfers to a non-Fourier lag estimator while failing under a shuffled-history control. Related work includes static sparse patterns, approximation-based attention, position extension, and dynamic prefill routing (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Guo et al., 2022; Kitaev et al., 2020; Wang et al., 2020; Choromanski et al., 2021; Xiong et al., 2021; Press et al., 2022; Chen et al., 2023; Ding et al., 2024; Jiang et al., 2024; Lai et al., 2025; Xu et al., 2025). KV selection, context extension, and serving systems address complementary streaming, decoding, or memory problems (Zhang et al., 2023; Xiao et al., 2024b; Tang et al., 2024; Li et al., 2024; Xiao et al., 2024a; Ratner et al., 2023; Yen et al., 2024; Ainslie et al., 2023; Kwon et al., 2023; Pope et al., 2023). Prior analyses identify recurring attention-head behaviors (Clark et al., 2019; Voita et al., 2019; Kovaleva et al., 2019; Michel et al., 2019); EchoStripe uses only preceding-layer routing evidence to make the current-layer causal decision.

2 Motivation

Let $\mathcal{S}_\ell^{\mathrm{cur}}$ denote the stripe set selected from current-layer evidence at layer ℓ, and $\mathcal{S}_\ell^{\mathrm{ref}}$ the set after Stripe Memory integration. The memory-added and memory-removed sets are

$$\mathcal{S}_\ell^{+} = \mathcal{S}_\ell^{\mathrm{ref}} \setminus \mathcal{S}_\ell^{\mathrm{cur}}, \qquad \mathcal{S}_\ell^{-} = \mathcal{S}_\ell^{\mathrm{cur}} \setminus \mathcal{S}_\ell^{\mathrm{ref}}. \tag{1}$$

For each layer–head pair, we compare the preceding-layer support and post-hoc subsequent-layer recurrence of $\mathcal{S}_\ell^{+}$ and $\mathcal{S}_\ell^{-}$; future layers never affect routing.

Figure 2 establishes why history should be treated as routing state rather than as an arbitrary perturbation. The effect is observed across samples and depth in Figure 2(a). More directly, Figure 2(b) compares the gain in preceding-layer consensus with the gain in subsequent-layer recurrence for each route revision: 75.9% of weighted revisions lie in the consensus–persistent quadrant. Thus, routes supported by history are more likely to remain useful beyond the current layer. Figure 2(c) shows that this structural signal can also translate into improved retrieval outcomes. Future-layer recurrence is used only for post-hoc validation; at inference time, EchoStripe accesses strictly preceding-layer profiles and current-layer evidence only. The controlled ablations in Tables 7 and 8 isolate the downstream contribution of this causal state.

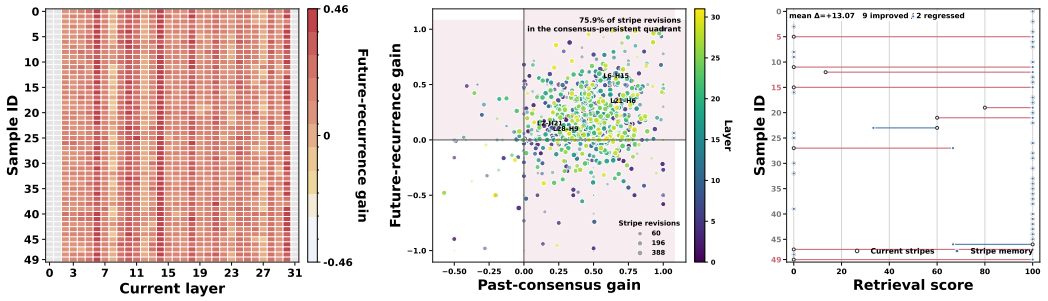


Figure 2: **Historical stripe evidence is predictive and useful.** **(a)** Future-recurrence gain across samples and depths. **(b)** Each point is a stripe revision: the horizontal axis measures the gain in strictly preceding-layer consensus and the vertical axis measures the post-hoc gain in subsequent-layer recurrence. Weighted by revision count, 75.9% of revisions lie in the positive-consensus, positive-recurrence quadrant. **(c)** Retrieval scores on illustrative inputs with current-layer routing alone and with Stripe Memory; the mean change is $+13.07$. Future layers are used only for offline validation, never for routing.

Table 1: **Routing-component boundary.** EchoStripe retains an efficient current-layer lag-evidence extractor, but changes routing from independent layer-wise selection to a causal, stateful cross-layer process.

Component	Prior routing	EchoStripe
Current-layer lag extraction	✓	✓
Independent layer-wise selection	✓	–
Recursive lag-profile state	–	✓
Head-specific reliability fusion	–	✓
Fixed-budget memory revision	–	✓
Token-guided mask correction	–	✓

3 ECHOSTRIPE ATTENTION

Consider a causal layer with sequence length L, block size B, and $N = \lceil L/B \rceil$. Block-mean pooled Q/K states $\bar{\mathbf{Q}}_h, \bar{\mathbf{K}}_h \in \mathbb{R}^{N \times d}$ construct the mask; retained blocks use original token-level states.

Figure 3 summarizes current-layer route construction, preceding-layer Stripe Memory, and final causal-mask assembly.

3.1 CURRENT-LAYER ROUTE CONSTRUCTION

For a causal block pair (i, j) with $j \le i$, let $b_{h,i,j} = \bar{\mathbf{q}}_{h,i}^{\mathsf{T}} \bar{\mathbf{k}}_{h,j} / \sqrt{d}$ be its mean-pooled QK score.

Directional routing and current lag evidence. At sampled causal locations and shifts $s \in \{1, 4\}$, EchoStripe compares score changes along a constant-lag diagonal with row- and column-direction changes:

$$
\begin{aligned}
e_{h,s}^{\text{diag}} &= \mathbb{E}_{(i,j)}[(b_{h,i+s,j+s} - b_{h,i,j})^2], \\
e_{h,s}^{\text{cross}} &= \tfrac{1}{2}\mathbb{E}_{(i,j)}[(b_{h,i+s,j} - b_{h,i,j})^2 + (b_{h,i,j+s} - b_{h,i,j})^2], \\
v_h &= \max_{s \in \{1,4\}} e_{h,s}^{\text{diag}} / (e_{h,s}^{\text{cross}} + \epsilon).
\end{aligned}
\tag{2}
$$

Heads with $v_h \le \theta_{\text{stripe}}$ and sufficient relative structural energy use the stripe route; all other heads use the complementary block route. Lower v_h indicates greater smoothness along constant-lag diagonals.

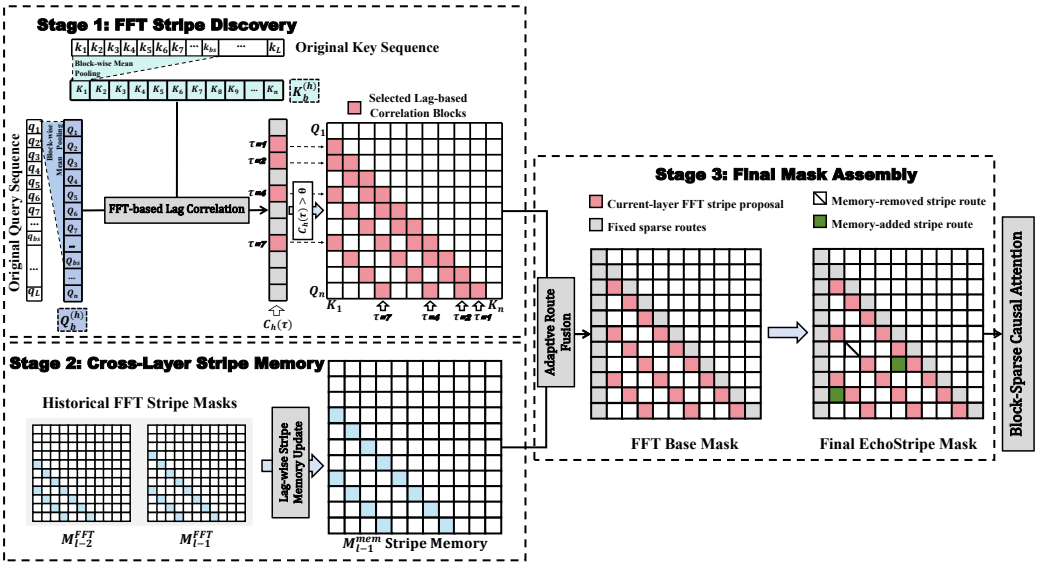


Figure 3: **EchoStripe Attention.** Current-layer route construction extracts lag evidence; Stripe Memory fuses it with a strictly preceding-layer lag-profile state; and Mask Refinement corrects the complete causal base mask. Pink, green, crossed, and gray blocks denote current, memory-added, memory-removed, and fixed routes, respectively.

For stripe heads, the current remote-lag profile is obtained by the frequency-domain cross-correlation used in prior lag routing (Anonymous, 2026):

$$\mathbf{c}_h = \mathrm{IFFT}\left(\sum_{r=1}^{d} \mathrm{FFT}(\widetilde{\mathbf{Q}}_{h,:,r}) \odot \overline{\mathrm{FFT}(\widetilde{\mathbf{K}}_{h,:,r})}\right), \qquad \mathcal{S}_h^{\mathrm{cur},(\ell)} = \mathrm{TopK}_{K_h^{(\ell)}}(\mathbf{s}_h^{(\ell)}). \tag{3}$$

Here $\mathbf{s}_h^{(\ell)}$ is the normalized causal portion of $\mathbf{c}_h$. Each selected lag τ retains its causal block diagonal $\mathcal{D}(\tau) = \{(i,j) : i - j = \tau,\ 0 \le j \le i < N\}$. Thus, the inherited component is only current-layer lag extraction; the stateful route integration and mask correction below are specific to EchoStripe.

Complementary and fixed routes. Non-stripe heads retain their allocated top-scoring causal blocks. For every head, EchoStripe additionally retains the allocated highest-scoring key columns, where $u_{h,j} = \sum_{i=j}^{N-1} b_{h,i,j}$ and $\mathcal{J}_h^{\mathrm{column}} = \mathrm{TopK}_{K_h^{\mathrm{column}}}(\{u_{h,j}\})$. The resulting column route is $\mathcal{M}_h^{\mathrm{column}} = \{(i,j) : j \le i,\ j \in \mathcal{J}_h^{\mathrm{column}}\}$. With sink span S, recent span R, and local span W, the fixed routes are $\mathcal{M}^{\mathrm{sink}} = \{(i,j) : j < S\}$, $\mathcal{M}^{\mathrm{recent}} = \{(i,j) : j \ge i - R\}$, and $\mathcal{M}^{\mathrm{local}} = \{(i,j) : i - W \le j \le i\}$, restricted throughout to causal pairs. The base mask is therefore

$$\mathcal{M}_h^{\mathrm{base}} = (\mathcal{M}_h^{\mathrm{route}} \cup \mathcal{M}_h^{\mathrm{column}} \cup \mathcal{M}^{\mathrm{sink}} \cup \mathcal{M}^{\mathrm{recent}} \cup \mathcal{M}^{\mathrm{local}}) \cap \mathcal{M}^{\mathrm{causal}}. \tag{4}$$

The fixed routes and the last-query-block fallback are protected from pruning.

3.2 Stripe Memory

Stage 2 stores only a one-dimensional, head-specific remote-lag profile $\mathbf{p}_h^{(\ell)}$, not token states or a two-dimensional mask. For a stripe head, it fuses this recursively propagated profile with the normalized current profile and preserves the current lag budget:

$$\widetilde{\mathbf{s}}_h^{(\ell)} = \alpha_h^{(\ell)} \mathbf{p}_h^{(\ell-1)} + (1 - \alpha_h^{(\ell)}) \mathbf{s}_h^{(\ell)}, \quad \mathcal{S}_h^{\mathrm{ref},(\ell)} = \mathrm{TopK}_{K_h^{(\ell)}}(\widetilde{\mathbf{s}}_h^{(\ell)}), \quad \mathbf{p}_h^{(\ell)} = \widetilde{\mathbf{s}}_h^{(\ell)}. \tag{5}$$

At the first layer, or without valid history, $\alpha_h^{(\ell)} = 0$. Non-stripe heads bypass this stage. Because the preceding profile already contains earlier fusions, this update propagates evidence recursively without storing every prior layer.

Adaptive memory weight. Let $\mathcal{T}_h^{(\ell-1)}$ and $\mathcal{T}_h^{(\ell-2)}$ be the top $K_{\text{hist}} = 8$ raw remote-lag sets at the preceding layers, and let $s_{(K_h)}$ and $s_{(K_h+1)}$ be current-profile scores at the actual TopK boundary. With $[\cdot]_0^1$ denoting clipping to $[0, 1]$, the implemented weight is

$$\begin{aligned} S_h &= |\mathcal{T}_h^{(\ell-1)} \cap \mathcal{T}_h^{(\ell-2)}|/K_{\text{hist}}, \quad C_h = \big[|s_{(K_h)} - s_{(K_h+1)}|/(|s_{(K_h)}| + \epsilon)\big]_0^1, \\ G_h &= [(d_{\text{high}} - v_h)/(d_{\text{high}} - d_{\text{low}})]_0^1, \quad A_h = S_h(1 - C_h), \\ \alpha_h^{(\ell)} &= \alpha_{\max} A_h + G_h\big[\alpha_{\min} + (\alpha_{\max} - \alpha_{\min})A_h - \alpha_{\max} A_h\big]. \end{aligned} \tag{6}$$

Invalid history sets $S_h = \alpha_h^{(\ell)} = 0$. We use $[\alpha_{\min}, \alpha_{\max}] = [0.65, 0.90]$ and $[d_{\text{low}}, d_{\text{high}}] = [0.80, 2.40]$: memory is emphasized only when preceding routes agree, the current decision is uncertain, and the head is stripe-like.

3.3 Mask Refinement

Stage 3 operates on the entire base mask and therefore remains defined when Stripe Memory is disabled. Let $\mathcal{M}_{h,i}^{\text{prot}}$ contain the fixed sink, recent, local, and fallback blocks for query block i. Writing $\text{TopK}_k(\mathcal{U}; b)$ for the k indices in set $\mathcal{U}$ with largest block scores $b_{h,i,j}$ (and analogously for BottomK), the compact candidate pool is

$$\mathcal{C}_{h,i} = \underbrace{\text{TopK}_8(\overline{\mathcal{M}}_{h,i}^{\text{base}}; b)}_{\mathcal{C}_{h,i}^{\text{add}}} \cup \underbrace{\text{BottomK}_8(\mathcal{M}_{h,i}^{\text{base}} \setminus \mathcal{M}_{h,i}^{\text{prot}}; b)}_{\mathcal{C}_{h,i}^{\text{prune}}} \cup \underbrace{\text{TopK}_4(\mathcal{M}_{h,i}^{\text{base}}; b)}_{\mathcal{C}_{h,i}^{\text{ref}}}, \tag{7}$$

where $\overline{\mathcal{M}}_{h,i}^{\text{base}}$ is the causal complement of the base mask. The pool has at most 20 candidates; each operator uses the smaller of its configured budget and the number of available candidates.

For valid query token t in block i and candidate block j, the token-level score, best selected reference, risk, and addition gain are

$$\begin{aligned} a_{h,t,j} &= \mathbf{q}_{h,t}^{\mathsf{T}} \bar{\mathbf{k}}_{h,j}/\sqrt{d}, \qquad \beta_{h,t} = \max_{j \in \mathcal{C}_{h,i} \cap \mathcal{M}_{h,i}^{\text{base}}} a_{h,t,j}, \\ r_{h,t} &= \max_{j \in \mathcal{C}_{h,i}^{\text{add}}} a_{h,t,j} - \beta_{h,t}, \qquad g_{h,i,j} = \max_{t \in \mathcal{T}_{h,i}} (a_{h,t,j} - \beta_{h,t}). \end{aligned} \tag{8}$$

Here $\mathcal{T}_{h,i}$ contains the highest-risk fraction $\rho = 0.40$ of valid tokens. Positive-gain candidates are ranked by $g_{h,i,j}$, and at most four blocks are added. With $\vartheta_{h,t}$ the 0.40 quantile of token t's scores over $\mathcal{C}_{h,i}$, pruning and final-mask construction are

$$\mathcal{P}_{h,i} = \{j \in \mathcal{C}_{h,i}^{\text{prune}} : a_{h,t,j} < \vartheta_{h,t}\ \forall t \in i\}, \qquad \mathcal{M}_h = [(\mathcal{M}_h^{\text{base}} \setminus \mathcal{P}_h) \cup \mathcal{A}_h \cup \mathcal{M}_h^{\text{protected}}] \cap \mathcal{M}^{\text{causal}}. \tag{9}$$

The unanimous-token criterion makes pruning conservative; $\mathcal{A}_h$ denotes accepted additions. A configured global keep cap is applied afterward.

3.4 Sparse Attention and Complexity

Sparse attention is evaluated only on retained token-level block pairs in $\mathcal{M}_h$. Let H be the number of heads. Non-stripe routing, column routing, and coarse refinement require $O(HN^2d)$ computation and $O(HN^2)$ block-score storage. FFT lag extraction adds $O(Nd \log N)$ time and $O(Nd)$ memory per stripe head; Stripe Memory adds $O(N)$ state. The fine stage uses at most $C = 20$ candidates per query block and costs $O(HNBCd)$. Thus, EchoStripe may be quadratic at the block level, but never materializes a token-level $L \times L$ attention matrix.

4 Experiments

4.1 Experimental Setup

Setup. We evaluate Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct on the 20-dataset LongBench suite and Llama-3.1-8B-Instruct on ten InfiniteBench tasks. All methods use identical instances, prompts, decoding, generation limits, and metrics; averages are unweighted across datasets.

Table 2: **Main results.** Full Attention is a dense reference. Best and second-best sparse-method scores are bold and underlined. Sparse methods use matched benchmark-level average realized keep ratios: 20% on LongBench and 12% on InfiniteBench.

(a) Performance on the common 20-dataset LongBench suite.

Method	Multi-document QA				Single-document QA				Summarization				Few-shot Learning				Synthetic Tasks		Code Completion		Avg.
	HotQA	2Wiki	MusiQ	DuRdr	MF-En	MF-Zh	NarQA	Qasp	GovR	QMSum	MNews	VCSum	TriQA	SAMSum	TREC	LSHT	PR-En	PCnt	LCC	RepoP	
Llama-3.1-8B-Instruct																					
Full Attention	32.93	21.90	12.28	28.92	40.81	53.51	35.34	34.91	35.14	23.03	26.25	17.07	91.05	47.71	84.00	40.00	98.00	6.18	50.92	41.04	41.05
FlexPrefill	14.47	17.05	11.70	27.32	32.81	44.99	27.89	23.06	31.78	22.29	25.62	16.81	88.84	47.44	72.00	30.00	30.20	4.18	46.56	**49.12**	33.21
MInference	**28.86**	**22.52**	11.96	28.22	40.73	51.05	**32.58**	32.81	**34.39**	**23.02**	26.37	**17.58**	87.05	**47.45**	82.00	40.00	**96.00**	**10.00**	**51.42**	41.94	**40.30**
XAttention	23.78	19.54	**12.51**	30.60	**45.35**	**55.42**	30.87	32.90	33.70	**23.02**	26.13	16.89	**94.16**	47.13	**84.00**	**43.00**	66.11	4.00	50.60	48.64	39.42
FFTSparse	25.30	20.54	10.71	30.10	40.54	49.93	28.74	34.23	33.86	22.34	26.29	16.89	91.05	46.75	82.00	40.00	60.67	4.98	48.28	35.27	37.42
EchoStripe	26.00	20.13	11.78	**31.06**	39.60	53.99	31.43	**34.47**	34.09	21.96	**26.48**	17.29	92.83	46.53	82.00	40.00	92.33	6.73	49.04	39.82	39.88
Qwen2.5-7B-Instruct																					
Full Attention	13.00	10.86	8.61	34.34	26.22	37.51	3.42	10.18	32.02	17.99	21.74	14.83	89.71	51.18	82.00	45.00	98.00	2.09	54.28	60.88	35.69
FlexPrefill	11.24	8.52	6.16	32.00	22.15	26.79	3.72	6.43	30.04	17.41	21.22	14.60	83.71	**52.57**	78.00	22.00	35.00	1.90	**59.88**	58.98	29.62
MInference	**13.41**	9.39	**9.41**	31.73	25.59	35.60	3.50	10.01	32.50	**17.98**	21.84	**15.27**	89.71	50.76	**86.00**	**43.00**	**94.33**	3.67	52.64	**65.10**	**35.57**
XAttention	11.26	11.24	7.83	33.62	25.45	33.30	3.79	9.58	**32.69**	**17.98**	21.64	14.67	**89.83**	51.78	84.00	38.00	80.83	1.65	54.60	61.44	34.26
FFTSparse	10.97	**11.40**	6.41	**34.63**	24.15	37.66	3.70	11.20	32.00	17.45	21.77	14.67	88.05	48.34	80.00	38.07	41.00	1.50	51.94	61.12	31.80
EchoStripe	11.65	10.15	7.37	34.11	**26.42**	**37.88**	**4.05**	**11.87**	32.22	17.15	**21.93**	15.02	**89.83**	50.98	82.00	39.20	80.00	**7.09**	51.38	64.16	34.72

(b) Performance on ten InfiniteBench tasks using Llama-3.1-8B-Instruct with 4-bit NF4 quantization.

Method	Code		Retrieval	Long-context QA				Synthetic and Mathematical Tasks			Avg.
	Code.D	Code.R	KV.R	LB.Cho	LB.Zh	LB.En	LD.En	MathF	NumStr	PassK	
Full Attention	28.00	2.00	20.00	50.00	9.86	15.29	24.00	26.00	98.00	100.00	37.32
FlexPrefill	28.00	**2.00**	10.00	38.00	8.28	10.25	**22.00**	32.00	**100.00**	**100.00**	35.05
MInference	**38.00**	**2.00**	**24.00**	38.00	8.80	10.50	20.00	**36.00**	98.00	**100.00**	**37.53**
XAttention	28.00	**2.00**	10.00	**44.00**	6.66	9.86	18.00	**36.00**	98.00	**100.00**	35.25
FFTSparse	26.00	0.00	10.00	34.00	6.78	7.86	**30.00**	32.00	92.00	94.00	33.26
EchoStripe	30.00	**2.00**	20.00	40.00	**8.99**	**10.67**	18.00	**36.00**	98.00	**100.00**	36.37

Baselines and matched sparsity. We compare Full Attention, FlexPrefill (Lai et al., 2025), MInference (Jiang et al., 2024), XAttention (Xu et al., 2025), FFTSparse (Anonymous, 2026), and EchoStripe. Sparse methods are independently calibrated to benchmark-level average realized causal-token-pair keep ratios of 20% on LongBench and 12% on InfiniteBench; Full Attention is a dense reference and excluded from sparse-method ranking.

Configuration. Unless otherwise specified, EchoStripe uses $B = 128$, sink, recent, and local ratios of 0.01, 0.01, and 0.02, QK and column-route ratios of 0.20 and 0.10, correlation and stripe thresholds of 0.85 and 2.00, $[\alpha_{\min}, \alpha_{\max}] = [0.65, 0.90]$, and $[d_{\text{low}}, d_{\text{high}}] = [0.80, 2.40]$. Mask Refinement uses addition, pruning, and reference budgets of 8, 8, and 4, respectively, a high-risk-token fraction of 0.40, at most four accepted additions per query-block row, and a pruning quantile of 0.40. Benchmark-specific controls and the full evaluation protocol are provided in Appendix A.

Efficiency protocol. We measure all 32 layers on an NVIDIA RTX 5090 at 16K–128K tokens. The operator includes selection, mask/layout construction, and attention, but excludes QKV projection, RoPE, output projection, MLP, transfer, and decoding.

4.2 Main Results

Table 2 shows that EchoStripe preserves near-best sparse-attention accuracy across backbones and task categories. Relative to MInference, the strongest sparse baseline in average accuracy, EchoStripe is lower by only 0.42 points on LongBench with Llama-3.1-8B-Instruct, 0.85 points on LongBench with Qwen2.5-7B-Instruct, and 1.16 points on InfiniteBench with Llama-3.1-8B-Instruct. These small differences should be interpreted jointly with the substantially different

Table 3: **Accuracy–efficiency summary.** Benchmark averages use the corresponding matched-realized-keep protocol. The 128K value is measured attention-operator latency over all 32 layers, not end-to-end generation latency.

Method	LB–Llama	LB–Qwen	Infinite	128K (ms)	vs. Full
MInference	40.30	35.57	37.53	14134.5	1.49×
XAttention	39.42	34.26	35.25	5184.0	4.05×
FFTSparse	37.42	31.80	33.26	2770.3	7.59×
EchoStripe	**39.88**	**34.72**	**36.37**	**2718.2**	**7.73×**

Table 4: **Matched-keep attention-operator latency.** Latency is measured in milliseconds and aggregates all 32 Transformer layers. Each entry reports *latency / speedup*. Sparse methods are independently tuned near the target realized causal token-pair keep shown in each column header. Speedup is relative to Full Attention at the same sequence length.

Method	16K (20%)	32K (15%)	64K (12%)	128K (10%)
Full Attention	348.0 / 1.00×	1345.9 / 1.00×	5292.1 / 1.00×	21016.5 / 1.00×
XAttention	**176.9 / 1.97**×	**452.2 / 2.98**×	1420.5 / 3.73×	5184.0 / 4.05×
FlexPrefill	553.5 / 0.63×	793.1 / 1.70×	1448.1 / 3.65×	3538.4 / 5.94×
MInference	1314.3 / 0.26×	2394.5 / 0.56×	5831.6 / 0.91×	14134.5 / 1.49×
FFTSparse	284.2 / 1.22×	491.6 / 2.74×	1069.7 / 4.95×	2770.3 / 7.59×
EchoStripe	336.5 / 1.03×	482.3 / 2.79×	**1028.9 / 5.14**×	**2718.2 / 7.73**×

operator-level computational costs reported below. Controlled ablations isolate the contribution of the proposed components.

Table 3 makes the operating point explicit. EchoStripe remains within 0.42–1.16 points of MInference across the three settings while requiring 2718.2 ms rather than 14134.5 ms for the measured 128K attention operator, a $5.20\times$ reduction in operator latency.

4.3 Efficiency

We use real news, meeting, and narrative texts and live post-RoPE Q/K states from all 32 layers.

At each length, method-specific controls are independently tuned near common realized causal-token-pair keep targets of 20%, 15%, 12%, and 10% at 16K, 32K, 64K, and 128K. The maximum deviation is 0.08 percentage points; this matches effective budgets without forcing a common mask structure.

This evaluation isolates the attention operator, including selection, mask/layout construction, and attention computation; it does not claim end-to-end serving speedup. EchoStripe becomes favorable from 32K onward and reaches $5.14\times$ and $7.73\times$ speedups over Full Attention at 64K and 128K, respectively.

At 128K, EchoStripe is $5.20\times$ faster than MInference at the measured attention-operator level while maintaining a $7.73\times$ speedup over Full Attention. Together with the 1.16-point InfiniteBench difference in Table 3, this result shows that the small accuracy gap to MInference should not be interpreted in isolation.

At 128K, current-layer lag discovery is the dominant selector cost; Stripe Memory and bounded Mask Refinement together account for 46.2%.

4.4 Ablation Study

The complete configuration achieves the best average (39.88). Removing Stripe Memory or Mask Refinement yields 37.42 and 36.74, whereas removing both yields 33.98, a 5.90-point decrease. Since every variant is matched to EchoStripe's realized keep within 0.05 percentage points, these differences cannot be attributed to a material change in the attention budget. Stripe Memory resolves uncertainty at the route level without enlarging the stripe-route budget, whereas Mask Refinement uses bounded token-level evidence to correct the complete causal base mask. Their complementary removals therefore degrade performance substantially.

Table 5: **Attention-operator latency breakdown at 128K.** All sparse methods are independently tuned to operate close to a 10% realized causal token-pair keep budget. Values are in milliseconds and aggregate all 32 Transformer layers. "Other" is the residual operator-level latency not attributed to the selector or attention kernel.

Method	Selector	Kernel	Other	Total	Speedup
Full Attention	0.0	21015.9	0.6	21016.5	1.00×
MInference	439.1	13618.9	76.5	14134.5	1.49×
XAttention	2691.9	**2446.6**	45.5	5184.0	4.05×
FlexPrefill	704.6	2831.9	1.9	3538.4	5.94×
FFTSparse	282.6	2455.0	32.7	2770.3	7.59×
EchoStripe	**252.2**	2448.0	18.0	**2718.2**	**7.73**×

Table 6: **Internal selector-stage latency of EchoStripe at 128K.** Results are averaged over multiple real 128K-token inputs and aggregate all 32 Transformer layers. All runs use the same approximately 10% realized causal token-pair keep setting.

Metric	FFT Discovery	Stripe Memory	Mask Refinement	Other	Total
Latency (ms)	127.9	54.2	62.2	7.9	**252.2**
Selector share	50.7%	21.5%	24.7%	3.1%	**100.0%**

4.4.1 Stripe Memory Mechanism Ablation

Previous-layer-only memory improves the average from 37.42 to 38.34, showing that aligned preceding-layer lag evidence is useful even without recursive history. Adaptive recursion further reaches 39.88. Fixed-α EMA (37.93) and History Voting (36.14) are weaker. Thus, the gain is not explained by injecting arbitrary history: effective cross-layer routing requires both recursive continuous state propagation and reliability-aware fusion, rather than a binary route cache, a fixed smoother, or an unweighted historical vote. Since all variants are matched to the same realized keep within 0.05 percentage points, the gain of Adaptive Recursive Memory reflects the memory rule rather than a larger attention budget.

4.4.2 Cross-Estimator Transfer and History Alignment

To test whether Stripe Memory is tied to frequency-domain lag extraction, we replace the FFT-based current-layer extractor with *DirectLag*, a non-Fourier alternative that directly scores causal pooled query–key block pairs along each relative-offset diagonal and aggregates this evidence into a lag profile. DirectLag produces the same one-dimensional profile interface as the default extractor; directional routing, deterministic base routes, the recursive memory state, and adaptive fusion are otherwise unchanged. To isolate cross-layer routing, Mask Refinement is disabled for every variant in this diagnostic. We evaluate all examples from six LongBench tasks: HotpotQA, Passage Retrieval EN, Passage Retrieval ZH, LCC, NarrativeQA, and Qasper. They cover multi-document reasoning, bilingual retrieval, long-document question answering, and code completion.

Table 9 shows that Stripe Memory transfers beyond frequency-domain correlation. With the non-Fourier DirectLag estimator, aligned Stripe Memory raises the average score from 32.65 to 37.72 (+5.07) at nearly unchanged realized keep (26.65% versus 26.84%). The shuffled-history control preserves the historical profile values and route budget but destroys their correspondence to relative-offset lags. It reduces the DirectLag average to 31.13 at a comparable 26.59% keep, a 6.59-point decrease from aligned Stripe Memory. Thus, the benefit is not explained by generic recursive smoothing, additional retained attention, or frequency-domain extraction; it depends on semantically aligned preceding-layer lag evidence.

5 Conclusion

EchoStripe formulates lag-structured sparse routing as a depth-aware causal process. Its Stripe Memory recursively maintains a compact one-dimensional lag-profile state and selectively fuses

Table 7: **Core-component ablation on LongBench.** All variants use Llama-3.1-8B-Instruct and the same 20 datasets. After variant-specific mask construction, unselected causal blocks are randomly added when necessary to match EchoStripe's realized causal token-pair keep of 20.12% within ± 0.05 percentage points. "w/o Both" is the current-layer-only EchoStripe variant.

Method	Multi-document QA				Single-document QA				Summarization				Few-shot Learning				Synthetic Tasks		Code Completion		Avg.	Keep (%)
	HotQA	2Wiki	MusiQ	DuRdr	MF-En	MF-Zh	NarQA	Qasp	Gov.R	QMSum	MNews	VCSum	TriQA	SAMSum	TREC	LSHT	PR-En	PCnt	LCC	RepoP		
EchoStripe (Full)	26.00	**20.13**	**11.78**	31.06	**39.60**	**53.99**	**31.43**	**34.47**	**34.09**	**21.96**	**26.48**	**17.29**	92.83	46.53	**82.00**	40.00	**92.33**	**6.73**	49.04	**39.82**	**39.88**	20.12 ± 0.05
w/o Stripe Memory	25.78	18.22	11.01	30.98	37.47	47.03	29.88	31.50	29.50	21.90	25.90	16.80	93.05	42.02	78.00	40.00	77.67	6.55	49.20	36.04	37.42	20.12 ± 0.05
w/o Mask Refinement	**29.42**	18.94	9.53	**31.59**	36.55	48.51	29.51	30.49	25.60	19.70	21.90	14.30	**95.05**	**46.58**	78.00	36.00	76.33	0.65	**50.30**	35.80	36.74	20.12 ± 0.05
w/o Both Components	24.24	17.35	9.04	27.72	35.02	46.56	27.78	28.43	23.80	18.20	20.10	13.40	85.05	41.89	76.00	**42.00**	63.27	2.47	45.46	31.80	33.98	20.12 ± 0.05

Table 8: **Stripe Memory mechanism ablation on LongBench.** All variants share FFT discovery, directional routing, complementary base-mask structures, and Mask Refinement; only the Stripe Memory rule differs. Unselected causal blocks are randomly added when necessary to match EchoStripe's 20.12% realized causal token-pair keep within ± 0.05 percentage points. Fixed-α EMA uses $\alpha = 0.75$.

Method	Multi-document QA				Single-document QA				Summarization				Few-shot Learning				Synthetic Tasks		Code Completion		Avg.	Keep (%)
	HotQA	2Wiki	MusiQ	DuRdr	MF-En	MF-Zh	NarQA	Qasp	Gov.R	QMSum	MNews	VCSum	TriQA	SAMSum	TREC	LSHT	PR-En	PCnt	LCC	RepoP		
Adaptive Recursive Memory	26.00	**20.13**	**11.78**	**31.06**	**39.60**	**53.99**	**31.43**	**34.47**	**34.09**	**21.96**	**26.48**	17.29	92.83	**46.53**	**82.00**	**40.00**	**92.33**	**6.73**	49.04	**39.82**	**39.88**	20.12 ± 0.05
Fixed-α EMA	**26.08**	18.13	10.98	30.25	37.65	48.52	30.28	32.54	31.89	21.06	26.12	**17.35**	**93.17**	43.26	78.65	39.00	81.36	6.32	48.89	37.16	37.93	20.12 ± 0.05
Previous-layer-only	24.50	16.80	9.50	30.80	38.20	50.10	30.50	33.20	32.50	21.50	26.30	17.10	92.90	44.80	79.50	39.50	85.00	6.60	48.95	38.50	38.34	20.12 ± 0.05
History Voting	23.80	15.50	8.80	29.50	36.00	45.00	28.50	30.00	29.00	20.50	25.00	16.50	92.50	41.00	77.00	38.00	76.00	6.20	48.50	35.50	36.14	20.12 ± 0.05
w/o Stripe Memory	25.78	18.22	11.01	30.98	37.47	47.03	29.88	31.50	29.50	21.90	25.90	16.80	93.05	42.02	78.00	**40.00**	77.67	6.55	**49.20**	36.04	37.42	20.12 ± 0.05

Table 9: **Cross-estimator transfer of Stripe Memory.** DirectLag is a non-Fourier current-layer lag estimator. "Shuffled" permutes the lag coordinates of the preceding memory profile while preserving its value distribution and route budget. Results aggregate all examples from the six listed LongBench tasks; Mask Refinement is disabled for every row.

Estimator	Memory	Alignment	Score	Keep
FFT correlation	None	–	33.29	26.29%
FFT correlation	Stripe Memory	Aligned	37.26	26.63%
DirectLag	None	–	32.65	26.65%
DirectLag	Stripe Memory	Aligned	**37.72**	26.84%
DirectLag	Stripe Memory	Shuffled	31.13	26.59%

historical and current evidence using an explicit head-specific reliability rule. Its Mask Refinement then translates this route-level decision into bounded token-guided corrections of the complete causal base mask. Across three benchmark/model settings, EchoStripe remains within 0.42–1.16 points of the strongest sparse baseline while reducing measured 128K attention-operator latency by $5.20\times$ relative to MInference and achieving a $7.73\times$ speedup over Full Attention. These results show that cross-layer routing evidence can recover much of the accuracy of substantially heavier dynamic sparse-attention methods without abandoning the efficiency of structured sparse routing. The cross-estimator and shuffled-history controls further show that this benefit is not tied to frequency-domain lag extraction or generic cross-layer smoothing, but depends on aligned preceding-layer lag evidence.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4890–4901. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.298/.

Anonymous. Anonymous submission, 2026. Under review.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 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. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.acl-long.172/.

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

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023. URL https://arxiv.org/abs/2306.15595.

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

Krzysztof M. Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286. Association for Computational Linguistics, 2019.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 11091–11104. PMLR, 2024. URL https://proceedings.mlr.press/v235/ding24i.html.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontañón, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. Longt5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 1073–1081. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.findings-naacl.55/.

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. URL https://arxiv.org/abs/2404.06654.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2407.02490.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4365–4374. Association for Computational Linguistics, 2019.

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

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2502.20766.

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

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In *Proceedings of Machine Learning and Systems*, volume 5, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/c4be71ab8d24cdfb45e3d06dbfca2780-Abstract-mlsys2023.html.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6383–6402. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.acl-long.352/.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2406.10774.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qVyeW-grC2k.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808. Association for Computational Linguistics, 2019.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. In *arXiv preprint arXiv:2006.04768*, 2020. URL https://arxiv.org/abs/2006.04768.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. URL https://arxiv.org/abs/2402.04617.

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

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14138–14148, 2021.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. Xattention: Block sparse attention with antidiagonal scoring. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.16428.

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.acl-long.142/.

Manzil Zaheer, Guru Guruganesh, Avinava Kumar Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.14048.

A Complete Implementation Details

This appendix records the formal EchoStripe configuration used for the reported benchmark experiments. Mask Refinement uses three candidate sets: unselected addition candidates, selected dynamic pruning candidates, and selected reference candidates. All candidate operations are causal; scores for a partially filled final query block use valid tokens only.

Table 10: Formal EchoStripe configuration for benchmark evaluation.

Group	Parameter	Value
Blocking	block size B	128
Current route	sink / recent / local ratios	0.01 / 0.01 / 0.02
Current route	QK / column route ratios	0.20 / 0.10
Current route	correlation / stripe thresholds	0.85 / 2.00
Current route	last-block threshold	0.01
Stripe Memory	$\alpha_{\min}$ / $\alpha_{\max}$	0.65 / 0.90
Stripe Memory	d_{low} / d_{high}	0.80 / 2.40
Stripe Memory	history TopK K_{hist}	8
Mask Refinement	add / prune / reference candidates	8 / 8 / 4
Mask Refinement	high-risk-token fraction ρ	0.40
Mask Refinement	maximum additions per query row	4
Mask Refinement	pruning quantile	0.40

B Benchmark and Matched-Keep Protocol

All LongBench results use the common 20-dataset suite, with Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct evaluated using identical instances, prompts, decoding settings, generation limits, and metrics across methods. InfiniteBench uses ten tasks with Llama-3.1-8B-Instruct in 4-bit NF4 quantization. Dataset averages are unweighted. Sparse methods are independently calibrated to benchmark-level average realized causal token-pair keep ratios of 20% on LongBench and 12% on InfiniteBench. Full Attention is a dense reference and is excluded from sparse-method ranking.

For the component and memory-mechanism ablations, every variant is matched to the full EchoStripe realized keep of 20.12% within ± 0.05 percentage points. When removing a component leaves causal capacity unused, causal-valid blocks are randomly filled after variant-specific mask construction. This matching changes the number of retained blocks but does not restore the removed module's routing signal; deterministic sink, recent, local, and fallback structures remain protected.

C Analysis Protocol for Historical Stripe Evidence

For the analysis in Figure 2, a route revision is a memory-added or memory-removed lag for a layer–head pair. Preceding consensus is computed only from available preceding-layer raw lag profiles. Subsequent recurrence is computed only after the routing decision as an offline diagnostic; no future-layer state is used to construct an inference-time mask. The 75.9% statistic in Figure 2(b) is weighted by the number of revisions represented by each plotted point. Figure 2(c) contains illustrative per-input retrieval cases and is not used to compute the benchmark averages.

D Detailed Results and Ablations

The complete per-dataset LongBench results for both backbones and the ten-task InfiniteBench results are reported in Table 2; they are kept in the main paper to permit direct task-level inspection. The full component and memory-mechanism ablations, including their matched realized keep values, are reported in Tables 7 and 8. The supplementary material contains per-dataset realized-keep, sample-completeness, and run-configuration records for every method.

E EFFICIENCY MEASUREMENT PROTOCOL

Efficiency is measured on an NVIDIA RTX 5090 using real news, meeting, and narrative inputs at 16K, 32K, 64K, and 128K tokens. All 32 Transformer layers use live post-RoPE Q/K states. Each sparse method is independently calibrated near realized causal token-pair keep targets of 20%, 15%, 12%, and 10%, respectively; the maximum reported target deviation is 0.08 percentage points. Latency is measured with CUDA events after warm-up and aggregates selection, mask/layout construction, and attention computation. It excludes model loading, tokenization, QKV projection, RoPE, output projection, MLP, CPU/GPU transfer, and generation/decode. Selector, kernel, and "Other" are defined as the selector stage, sparse attention kernel, and residual operator-level time, respectively. Consequently, all reported speedups are attention-operator speedups rather than end-to-end serving speedups.