RobinAttention: Dynamic Block Sparse Attention via Per-Head Round-Robin Shifts for Long-Context Inference

Anonymous ACL submission

Abstract

The quadratic complexity of attention mechanisms poses a critical bottleneck for large language models processing long contexts. While dynamic sparse attention methods offer input-adaptive efficiency, they face fundamental trade-offs: requiring preprocessing, lacking global evaluation, violating query independence, or incurring high computational overhead. We present RobinAttention, a novel dynamic sparse attention method that simultaneously achieves all desirable properties through a head-robin sampling strategy. By rotating query sampling positions across attention heads within each stride, RobinAttention maintains query independence while enabling efficient global pattern discovery with stride-level aggregation. Our method reduces complexity from $O(L^2)$ to $O(L^2/S^2)$ and employs adaptive Top-τ selection for optimal sparsity. Extensive experiments on natural language understanding (HELMET) and multimodal video comprehension (Video-MME) demonstrate that RobinAttention recovers over 99% of full attention performance while computing only half of the attention blocks, achieving 2.4× speedup at 128K context length and consistently outperforming existing dynamic sparse attention methods.

1 Introduction

The ability to process long contexts has become increasingly critical for large language models (LLMs), enabling applications such as multi-document reasoning and code repository analysis. Recent advances (Chen et al., 2023; Peng et al., 2024; Liu et al., 2023; Wu et al., 2025; Baidu-ERNIE-Team, 2025) have pushed context windows to 128K tokens and beyond, significantly expanding the scope of addressable problems. However, the quadratic computational complexity $O(L^2)$ of the attention mechanism (Ashish, 2017) poses a fundamental bottleneck, making long-context inference prohibitively expensive for deployment.

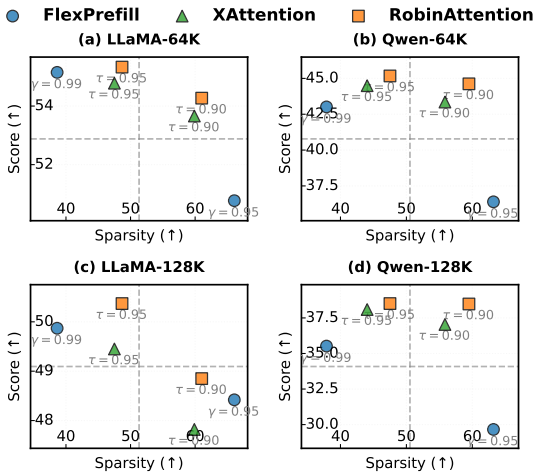

Figure 1: Sparsity-accuracy trade-offs across different models and context lengths.

Sparse attention has emerged as a promising solution by selectively computing only the most important attention scores. Existing approaches can be broadly categorized into static methods that use predetermined sparsity patterns, and dynamic methods that adaptively determine sparse patterns based on input characteristics. While static methods like BigBird (Zaheer et al., 2020) and StreamingLLM (Xiao et al., 2023) achieve computational efficiency, their fixed patterns cannot adapt to varying attention distributions, potentially missing critical dependencies. Dynamic methods (Gao et al., 2024; Jiang et al., 2024; Lai et al., 2025; Xu et al., 2025) provide input-adaptive sparsity but face different trade-offs in design choices that affect their practical effectiveness.

Through systematic analysis of existing dynamic sparse attention methods, we identify five critical dimensions that characterize their effectiveness: (i) preprocessing requirements that affect deployment flexibility, (ii) global evaluation capability for capturing long-range dependencies, (iii) query independence for maintaining semantic integrity, (iv) pattern flexibility across different attention heads,

and (v) softmax granularity that impacts computational efficiency. Current methods excel in some dimensions while compromising others—no existing approach simultaneously achieves all desirable properties, leading to suboptimal trade-offs between accuracy and efficiency.

We present RobinAttention, a novel dynamic sparse attention method that achieves an optimal balance across all critical dimensions through a head-robin sampling strategy. Our key insight is that by rotating query sampling positions across attention heads within each stride, we can maintain query independence while enabling efficient global pattern discovery. Specifically, RobinAttention: (i) samples one representative query per stride using a round-robin scheme across heads; (ii) performs stride-level aggregation that reduces computational cost from $O(L^2)$ to $O(L^2/S^2)$; and (iii) employs adaptive Top-τ block selection based on attention concentration patterns.

Extensive experiments demonstrate RobinAttention's effectiveness across comprehensive benchmarks in natural language understanding and multimodal video comprehension. Evaluating on HELMET and Video-MME, RobinAttention recovers over 99% of full attention performance while computing only half of the attention blocks, significantly outperforming both FlexPrefill and XAttention across various sparsity settings. Furthermore, RobinAttention achieves 2.4× end-to-end speedup at 128K context length and reduces pattern search overhead by 18.2% compared to XAttention, making it both the fastest and most accurate sparse attention method in our evaluation.

The main contributions of this paper are:

- We provide a systematic characterization of dynamic sparse attention methods across five critical dimensions, revealing fundamental trade-offs in existing approaches.
- We introduce RobinAttention, a novel head-robin sampling strategy that uniquely achieves all desirable properties: preprocessing-free deployment, global evaluation, query independence, pattern-agnostic simplicity, and efficient stride-level computation.
- Comprehensive experiments on both language (HELMET) and multimodal (Video-MME) benchmarks validate our approach's superiority in balancing efficiency and accuracy across diverse long-context scenarios.

2 Preliminary and Background

2.1 Block-wise Sparse Attention

For an input sequence of length L, the query matrix $\mathbf{Q} \in \mathbb{R}^{L\times d}$ and key matrix $\mathbf{K} \in \mathbb{R}^{L\times d}$ are partitioned into consecutive blocks of size B, where d denotes the attention head dimension. Let $N_b = \lceil L/B \rceil$ be the total number of blocks. The i-th query block and j-th key block are denoted as $\mathbf{Q}^{(i)}$ and $\mathbf{K}^{(j)}$ respectively, where $i, j \in \{1, 2, \ldots, N_b\}$.

In block-wise sparse attention (Guo et al., 2024), a block selection matrix $\mathbf{B} \in \{0, 1\}^{N_b\times N_b}$ is defined, where $\mathbf{B}_{ij} = 1$ indicates that attention scores are computed between the i-th query block and the j-th key block, while $\mathbf{B}_{ij} = 0$ indicates that the computation for this block pair is skipped. The sparse attention computation is formalized as:

$$\mathbf{A}_{\text{sparse}} = \text{Softmax}\left(\frac{1}{\sqrt{d}}\mathbf{Q}\mathbf{K}^T \odot \mathbf{M}_{\mathbf{B}}\right)\mathbf{V} \quad (1)$$

where $\mathbf{M}_{\mathbf{B}} \in \mathbb{R}^{L\times L}$ is the fine-grained mask matrix expanded from the block selection matrix $\mathbf{B}$:

$$\mathbf{M}_{\mathbf{B}}[i, j] = \begin{cases} 1, & \text{if } \mathbf{B}_{\lfloor i/B\rfloor,\lfloor j/B\rfloor} = 1 \text{ and } j \le i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The optimization objective seeks to minimize the approximation error between sparse and full attention outputs while constraining computational cost:

$$\min_{\mathbf{B}} \quad \|\mathbf{A}_{\text{full}} - \mathbf{A}_{\text{sparse}}\|_F \quad (3)$$

$$\text{s.t.} \quad \sum_{i,j} \mathbf{B}_{ij} \le \rho \cdot N_b^2 \quad (4)$$

$$\mathbf{B}_{ij} = 0, \quad \forall j > i \quad (5)$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\rho \in (0, 1]$ is a sparsity parameter, and the second constraint ensures causality for autoregressive models.

2.2 Multi-Dimensional Analysis of Attention Selection

Sparse attention mechanisms can be categorized into static and dynamic patterns. Among static sparse methods, approaches such as BigBird (Zaheer et al., 2020), StreamingLLM (Xiao et al., 2023), and TriangleMix (He et al., 2025) employ a predetermined block selection matrix $\mathbf{B}_{\text{static}}$ that remains constant regardless of the input sequence, effectively reducing computational complexity. However, the dynamic nature of attention sparsity has

Table 1: Comparison of dynamic sparse attention methods

Method	Preprocessing -free	Global Evaluation	Query Independence	Pattern -agnostic	Softmax Granularity
SeerAttention	✗	✓	✓	✓	Block
MInference	✗	✗	✓	✗	Token
FlexPrefill	✓	✗	✓	✗	Token
XAttention	✓	✓	✗	✓	Stride
RobinAttention	✓	✓	✓	✓	Stride

been well-documented in recent works, demonstrating that optimal attention patterns vary significantly across different inputs and layers. Static patterns cannot adapt $\mathbf{B}$ based on input-specific characteristics of $\mathbf{Q}$ and $\mathbf{K}$, limiting their ability to capture critical dependencies.

Dynamic sparse attention methods adaptively determine the block selection matrix $\mathbf{B}_{dynamic}$ based on input-specific characteristics, enabling more flexible and accurate sparse patterns compared to static approaches. We summarize the key characteristics of existing dynamic sparse attention methods in Table 1.

SeerAttention (Gao et al., 2024) employs block-wise pooling of $\mathbf{Q}$ and $\mathbf{K}$ matrices followed by learned linear transformations through an AttnGate module trained via distillation. **MInference** (Jiang et al., 2024) offline assigns optimal sparse patterns to each attention head, then dynamically constructs sparse indices during inference using the last query block for vertical-slash heads or mean pooling for block-sparse heads. **FlexPrefill** (Lai et al., 2025) also leverages the last query block for pattern discovery but adaptively determines query-aware or vertical-slash patterns using Jensen-Shannon divergence as a reliability metric. **XAttention** (Xu et al., 2025) samples anti-diagonal elements at stride granularity and achieves summation through dimension aggregation to estimate importance across the entire attention matrix. These methods exhibit distinct trade-offs across the following dimensions:

- *Preprocessing-free* indicates whether a method requires offline training or pattern search. Methods requiring preprocessing (SeerAttention's distillation training, MInference's offline pattern assignment) achieve better accuracy but limit deployment flexibility, while preprocessing-free methods (FlexPrefill, XAttention) offer immediate applicability to new models and tasks.

- *Global evaluation* determines whether importance estimation considers the entire query-key space. Global methods (SeerAttention, XAttention) capture long-range dependencies more effectively but incur higher computational overhead, whereas local methods (MInference, FlexPrefill) that only use the last query block are more efficient but may miss important patterns in earlier positions.

- *Query independence* ensures each query's attention distribution is computed separately without cross-query interference. Methods violating this principle (XAttention's anti-diagonal aggregation across different queries) risk mixing attention distributions from different contexts, potentially introducing evaluation errors, while query-independent methods preserve the semantic integrity of each query's attention pattern.

- *Pattern-agnostic* reflects whether a method applies a unified strategy across all attention heads. Pattern-specific methods (MInference, FlexPrefill) achieve better accuracy by tailoring to different attention patterns but require additional pattern detection overhead, while pattern-agnostic approaches (SeerAttention, XAttention) offer simpler implementation and consistent behavior.

- *Softmax granularity* significantly impacts computational efficiency: block-level or stride-level aggregation (SeerAttention, XAttention) accelerates pattern discovery through dimension reduction but may lose fine-grained information, while token-level softmax (MInference, FlexPrefill) preserves precise attention distributions at the cost of higher computational complexity.

RobinAttention presents a balanced solution that

combines the strengths of existing approaches: it requires no preprocessing for maximum flexibility, performs global evaluation for comprehensive pattern capture, maintains query independence for accurate attention computation, applies a pattern-agnostic strategy for simplicity, and uses stride-level softmax for computational efficiency. This design achieves an optimal trade-off between accuracy and efficiency for dynamic sparse attention in long-context scenarios.

3 Methodology: RobinAttention

RobinAttention achieves efficient dynamic sparse attention through a novel head-robin sampling strategy that maintains query independence while enabling fast pattern discovery. Figure 2 illustrates the complete workflow of RobinAttention.

3.1 Query Sampling with Head-Robin Strategy

To maintain query independence, RobinAttention samples one representative query per stride S tokens to compute attention with all keys, ensuring that each query's attention distribution is computed separately without cross-query interference. This preserves the semantic integrity of each query's attention pattern. For attention head h, the sampled query position $P(i, h)$ within each stride interval $[iS, (i+1)S)$ is determined by:

$$P(i, h) = iS + (S - 1 - (h \bmod S)) \quad (6)$$

The head-robin strategy serves two critical purposes: (i) it ensures that across different heads, all positions within a stride are eventually sampled, preventing information loss that would occur with fixed-position sampling; (ii) the varied sampling positions naturally intersect with important vertical and slash patterns in the attention matrix, enabling effective detection of these critical structures. The sampled query set for head h is:

$$\mathbf{Q}_{\text{sample}}^{(h)} = \{\mathbf{Q}_j : j = P(i, h), i \in [0, \lfloor L/S \rfloor)\} \quad (7)$$

3.2 Stride-level Importance Estimation

To accelerate pattern discovery, we aggregate attention scores at stride granularity through dimension reduction. For query stride i and key stride j, we compute an importance score representing their aggregated attention by first aggregating key vectors within each stride and then computing the dot product with the sampled query:

$$\mathbf{I}_{i,j}^{(h)} = \frac{1}{S\sqrt{d}} \mathbf{Q}_{P(i,h)} \cdot \left(\sum_{k=jS}^{(j+1)S-1} \mathbf{K}_k \right) \quad (8)$$

where $\mathbf{Q}_{P(i,h)}$ is the query vector at the sampled position from stride i for head h, and the summation aggregates all key vectors within stride j. The normalization factor $\frac{1}{S\sqrt{d}}$ accounts for both attention scaling and stride size.

We then apply row-wise softmax to obtain normalized importance scores for each query stride:

$$\mathbf{P}_{i,j}^{(h)} = \frac{\exp(\mathbf{I}_{i,j}^{(h)})}{\sum_{j'=0}^{\lfloor L/S \rfloor - 1} \exp(\mathbf{I}_{i,j'}^{(h)})} \quad (9)$$

Through query sampling and key aggregation, this stride-level importance estimation reduces the computational complexity from $O(L^2)$ to $O(L^2/S^2)$, while maintaining global evaluation capability as every position in the sequence contributes through our head-robin sampling strategy.

3.3 Block-level Selection via Top-τ Thresholding

To align with our block-wise sparse attention formulation and enable efficient GPU computation, we aggregate stride-level importance scores to block level. For query block m and key block n, the block importance score is computed as:

$$\mathbf{S}_{m,n}^{(h)} = \sum_{\substack{i \in \text{block } m \\ j \in \text{block } n}} \mathbf{P}_{i,j}^{(h)} \quad (10)$$

For each query block m, we employ a top-τ selection strategy that retains blocks whose cumulative importance exceeds threshold τ. Specifically, we sort key blocks by importance and select the minimal set such that:

$$\mathbf{B}_{m,n}^{\text{dynamic}} = \begin{cases} 1, & \text{if } n \in \{k : \sum_{l \in \text{sort}(\mathbf{S}_{m,*}^{(h)})[1:k]} \mathbf{S}_{m,l}^{(h)} \geq \tau\} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Following prior work demonstrating the critical importance of final query positions for maintaining generation quality, we construct a protected mask $\mathbf{B}^{\text{static}}$ that unconditionally retains all blocks for the final query block. The final block selection matrix combines both components:

$$\mathbf{B}_{m,n} = \mathbf{B}_{m,n}^{\text{dynamic}} \vee \mathbf{B}_{m,n}^{\text{static}} \quad (12)$$

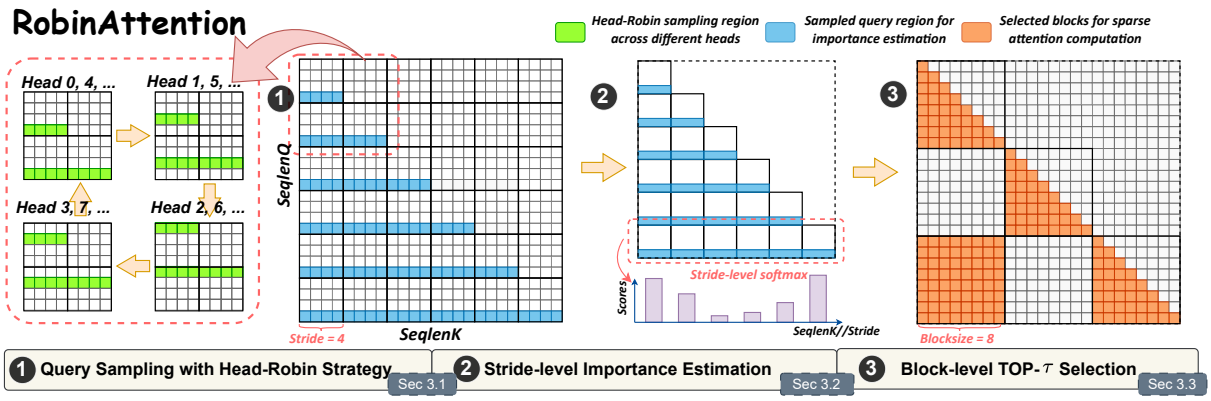

Figure 2: Illustration of RobinAttention. ①, ②, and ③ represent the three stages of our method. The example shows a configuration with stride size $S = 4$ and block size $B = 8$.

4 Experiments

4.1 Settings

Models. For natural language tasks, we evaluate two prominent long-context language models: (i) **Meta-LLaMA-3.1-8B-Instruct** (Grattafiori et al., 2024), which supports up to 128K tokens context length with strong instruction-following capabilities; (ii) **Qwen2.5-7B-Instruct** (Qwen et al., 2025), which supports up to 128K tokens context length through YARN (Peng et al., 2023) scaling with enhanced multilingual performance. For video understanding tasks, we evaluate **Qwen2-VL-7B-Instruct** (Wang et al., 2024), a vision-language model capable of processing and understanding video content through multimodal attention. We maintain the default chat template settings for all test datasets to ensure consistent evaluation across different models.

Baselines. We evaluate several state-of-the-art baseline methods: (i) **FlashAttention** (Dao et al., 2022; Dao, 2024), a highly optimized attention kernel that serves as our dense baseline for measuring the performance of standard full attention computation. (ii) **FlexPrefill**, a dynamic sparse attention method that captures the Vertical-Slash pattern through the last query block; we test it with hyperparameter $\tau = 0.1$ and two threshold settings $\gamma = 0.95$, and $\gamma = 0.99$. (iii) **XAttention**, a dynamic method that identifies important block patterns through anti-diagonal evaluation; we evaluate it with block size $S = 8$ and two threshold settings $\tau = 0.9$ and $\tau = 0.95$. (iv) **RobinAttention**, our proposed method that employs head-robin sampling for global pattern discovery; we evaluate it with the same configurations as XAttention. All sparse attention experiments focus on the prefill stage, while the decoding stage employs uniform dense computation for all methods.

Datasets. We evaluate on the following benchmarks across two task categories: (i) For natural language understanding tasks, we employ **HELMET** (Yen et al., 2025), a comprehensive long-context benchmark that includes seven task categories: retrieval-augmented generation (Natural Questions, TriviaQA, HotpotQA), generation with citations (ALCE), passage re-ranking (MS MARCO), many-shot in-context learning (TREC, BANKING77, CLINC150), long-document QA (NarrativeQA, ∞Bench), summarization (MultiLexSum), and synthetic recall (RULER, JSON KV). (ii) For video understanding tasks, we use **Video-MME** (Fu et al., 2024), the first full-spectrum multimodal evaluation benchmark for MLLMs, containing 900 videos across three duration categories with 2700 multiple-choice QA pairs. The dataset requires models to comprehend temporal video content across 12 task types, including perception, reasoning, and information synthesis. All experiments are conducted on NVIDIA H100 GPUs.

4.2 Main Result

HELMET. Table 2 and Figure 1 present the average performance across all HELMET tasks at context lengths from 8K to 128K tokens. RobinAttention consistently achieves the highest accuracy among all sparse attention methods on both models. At the conservative setting ($\tau = 0.95$ and $\gamma = 0.99$), RobinAttention achieves 56.24 on Llama, compared to XAttention's 55.74 and FlexPrefill's 56.02 (48.68% vs 47.50% and 38.63% sparsity). On Qwen with the same settings, RobinAttention achieves 47.17, outperforming XAtten-

Table 2: Performance comparison on HELMET benchmark across different context lengths.

Model	Methods	Avg. Sparsity(↑)	8K	16K	32K	64K	128K	Avg.(↑)
LLaMA	FullAttention	- (0%)	61.17	58.67	56.71	55.81	49.74	56.42
	FlexPrefill	0.95 (66.07%)	56.02	54.48	53.01	50.77	48.42	52.54
		0.99 (38.63%)	60.73	58.15	56.22	55.15	49.87	56.02
	XAttention	0.90 (59.88%)	59.92	57.15	55.54	53.66	47.82	54.82
		0.95 (47.50%)	60.79	58.10	55.57	54.78	49.45	55.74
	RobinAttention	0.90 (61.02%)	**60.95**	**58.43**	**55.86**	**54.27**	**48.85**	**55.67**
		0.95 (48.68%)	**61.22**	**58.21**	**56.07**	**55.33**	**50.37**	**56.24**
Qwen	FullAttention	- (0%)	54.75	50.66	48.40	44.98	39.59	47.68
	FlexPrefill	0.95 (63.21%)	44.11	40.59	40.60	36.39	29.66	38.27
		0.99 (37.88%)	53.18	49.05	46.48	43.01	35.52	45.45
	XAttention	0.90 (55.85%)	52.53	47.72	45.72	43.35	37.04	45.27
		0.95 (44.05%)	53.90	49.77	46.96	44.49	38.09	46.64
	RobinAttention	0.90 (57.02%)	**53.37**	**49.31**	**47.35**	**44.62**	**38.48**	**46.63**
		0.95 (44.68%)	**54.22**	**50.02**	**47.92**	**45.17**	**38.51**	**47.17**

Table 3: Performance comparison on Video-MME benchmark using Qwen2-VL-7B-Instruct.

FPS	Method	Avg. Sparsity(↑)	Short	Medium	Long	Avg.(↑)
1fps	FullAttention	- (0%)	72.90	63.60	55.20	63.90
	FlexPrefill	0.95 (46.70%)	**72.80**	62.00	54.00	62.90
		0.99 (17.80%)	72.40	63.60	55.80	63.90
	XAttention	0.90 (49.30%)	71.80	62.80	55.70	63.40
		0.95 (37.50%)	**72.70**	63.60	56.10	64.10
	RobinAttention	0.90 (47.00%)	72.20	**63.60**	**56.20**	**64.00**
		0.95 (34.70%)	72.60	**64.00**	**56.20**	**64.30**
0.5fps	FullAttention	- (0%)	72.10	68.80	62.40	67.80
	FlexPrefill	0.95 (58.60%)	70.80	67.00	59.40	65.70
		0.99 (28.80%)	71.70	68.40	61.90	67.30
	XAttention	0.90 (51.50%)	**72.00**	68.00	61.10	67.00
		0.95 (39.90%)	**72.30**	**68.90**	**62.30**	**67.90**
	RobinAttention	0.90 (49.30%)	71.70	**68.70**	**62.20**	**67.50**
		0.95 (37.40%)	72.20	68.70	**62.30**	67.70

tion's 46.64 and FlexPrefill's 45.45 (44.68% vs 44.05% and 37.88% sparsity). Notably, RobinAttention recovers 99.6% and 99.0% of FullAttention performance on Llama and Qwen respectively, while computing approximately half of the attention blocks.

The accuracy-sparsity trade-off advantage of RobinAttention is even more pronounced at the aggressive setting ($\tau = 0.90$ and $\gamma = 0.95$). On Llama, RobinAttention maintains 55.67, significantly outperforming XAttention's 54.82 and FlexPrefill's 52.54 (61.02% vs 59.88% and 66.07% sparsity). The pattern holds on Qwen, where RobinAttention achieves 46.63, compared to XAttention's 45.27 and FlexPrefill's 38.27 (57.02% vs 55.85% and 63.21% sparsity). This demonstrates that RobinAttention more effectively identifies critical attention patterns—achieving higher sparsity than XAttention while maintaining better accuracy, and preserving significantly better accuracy than FlexPrefill despite its higher sparsity. Detailed analysis of performance on all individual subtasks in HELMET is provided in Appendix A.

Video-MME. Table 3 presents results on the Video-MME benchmark. At the standard 1fps setting, RobinAttention consistently achieves the highest overall accuracy under both conservative and aggressive configurations, outperforming XAttention and FlexPrefill while maintaining comparable or better sparsity levels. This advantage becomes particularly pronounced on Medium and Long videos. Unlike text-based tasks where attention patterns often concentrate on specific tokens, video understanding requires modeling complex spatiotempo-

ral relationships distributed across frames, making RobinAttention's global evaluation capability especially valuable for capturing temporal dependencies in extended sequences. At the lower 0.5fps sampling rate, RobinAttention maintains competitive performance across settings. These results validate our method's effectiveness in multimodal scenarios with diverse attention characteristics.

Runtime Efficiency. Figure 3 presents the runtime comparison on Meta-Llama-3.1-8B-Instruct across context lengths from 8K to 128K tokens. At 128K context length, RobinAttention achieves a 2.4× speedup over Full Attention while delivering the fastest inference speed among all sparse attention methods. This efficiency advantage is particularly valuable given that RobinAttention simultaneously achieves state-of-the-art accuracy on HELMET benchmark, as demonstrated in Table 2.

The superior efficiency is further evidenced by pattern search overhead. As shown in Figure 3(b), our head-robin sampling strategy combined with stride-level aggregation reduces pattern discovery time by 18.2% compared to XAttention at 128K tokens. This improvement stems from our design that avoids redundant computation through systematic query sampling and efficient importance estimation, demonstrating RobinAttention's effectiveness for practical deployment in long-context scenarios.

4.3 Ablation Study

Final Query Block Protection. To validate the effectiveness of protecting the final query block, we apply this strategy to XAttention baseline. As shown in Table 5, adding final block protection to XAttention yields modest improvements (from 55.74 to 55.92) with sparsity cost, confirming its utility for maintaining generation quality. However, RobinAttention achieves substantially better performance (56.24), demonstrating that effective pattern discovery through our robin strategy contributes more significantly to overall performance than the protection mechanism alone.

Robin Strategy. To isolate the impact of robin scheduling, we compare performance without robin mechanism and across three robin strategies: head-robin, layer-robin, and hybrid-robin. As shown in Table 6, while all three strategies achieve comparable performance, head-robin consistently outperforms alternatives across most context lengths, demonstrating superior effectiveness in capturing critical attention patterns. These results validate that diversifying query positions across attention heads provides the most effective approach for comprehensive pattern coverage, combining both simplicity and robust performance.

Stride Sizes. The stride size S determines the granularity of query sampling in RobinAttention, directly influencing the balance between pattern estimation accuracy and computational efficiency. As shown in Table 4, for stride configurations $S < 32$, RobinAttention maintains stable performance with minimal accuracy variation and consistently high sparsity. However, at larger stride $S = 32$, coarse-grained aggregation dilutes important signals from critical tokens, leading to accuracy drops. These results demonstrate that the head-robin sampling strategy effectively captures critical attention patterns across moderate stride ranges without requiring excessively fine-grained query sampling, thereby achieving both robust pattern estimation and computational efficiency. Compared to XAttention baseline, RobinAttention consistently achieves superior or comparable accuracy with higher sparsity across all stride settings.

Block Selection Accuracy. We validate RobinAttention's importance estimation through post-hoc comparison with full attention ground truth. As detailed in Appendix B, RobinAttention achieves higher precision and F1 scores, confirming more accurate important block identification.

5 Related Work

5.1 Training-based Sparse Methods

Several approaches integrate sparsity into the training process. NSA (Yuan et al., 2025) learns sparse regions during training, while MOBA (Lu et al., 2025) employs mixture-of-experts for sparse computation. These methods achieve strong performance but require training or fine-tuning, limiting applicability to pre-trained models.

5.2 Inference-oriented Sparse Methods

Static Sparse Patterns. Static methods (Zaheer et al., 2020; Beltagy et al., 2020; Xiao et al., 2023; He et al., 2025) use predetermined patterns such as local windows, sink tokens, and triangular masks, but cannot adapt to input-specific characteristics.

Dynamic Sparse Patterns. Dynamic methods adaptively compute attention masks during inference. SeerAttention (Gao et al., 2024) employs learned pooling for block selection, MInference (Jiang et al., 2024) assigns patterns based on offline profiling, FlexPrefill (Lai et al., 2025) uses

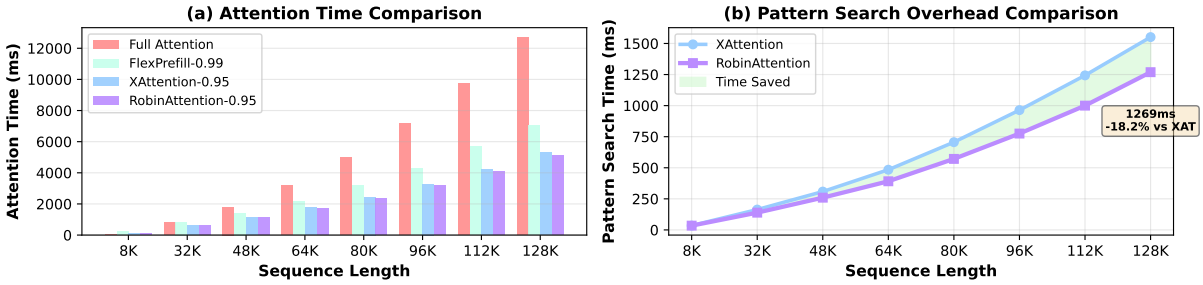

Figure 3: Runtime comparison of attention methods on LLaMA-3.1-8B-Instruct across different context lengths. (a): Attention overhead. (b): Pattern search time.

Table 4: Performance comparison of different strides on HELMET using LLaMA-3.1-8B-Instruct.

Method	S=4		S=8		S=16		S=32	
	Score	Sparsity	Score	Sparsity	Score	Sparsity	Score	Sparsity
XAttention-0.90	55.24	60.50%	54.82	60.08%	54.56	59.43%	54.50	57.69%
RobinAttention-0.90	**55.62**	**61.38%**	**55.67**	**61.02%**	**55.37**	**60.70%**	**54.97**	**60.02%**
XAttention-0.95	**55.89**	48.23%	55.74	47.50%	55.45	46.46%	55.49	45.27%
RobinAttention-0.95	55.85	**49.03%**	**56.24**	**48.68%**	**55.91**	**47.96%**	**55.62**	**47.78%**

Table 5: Ablation study on the impact of protecting the final query block on HELMET use LLaMA-3.1-8B-Instruct.

Method	Sparsity	Sink & Recent Block	Final Block	8K	16K	32K	64K	128K	Avg.
XAttention-0.95	47.50%	✓	✗	60.79	58.10	55.57	54.78	49.45	55.74
XAttention+B^{static}	46.40%	✓	✓	61.09	**58.27**	55.94	54.51	49.77	55.92
RobinAttention-0.95	49.06%	✓	✓	**61.44**	58.19	55.74	**55.51**	50.00	56.18
RobinAttention-0.95	48.68%	✗	✓	61.22	58.21	**56.07**	55.33	**50.37**	**56.24**

Table 6: Isolated ablation study on Robin scheduling strategies on HELMET use LLaMA-3.1-8B-Instruct.

Method	8K	16K	32K	64K	128K
No Robin	60.91	57.86	55.67	54.62	49.18
Head-Robin	**60.93**	57.90	**56.03**	54.70	**49.44**
Layer-Robin	60.75	57.60	55.99	**54.95**	48.40
Hybrid-Robin	60.42	**57.96**	56.03	54.54	49.10

divergence metrics for pattern discovery, and XAttention (Xu et al., 2025) performs anti-diagonal query sampling with stride aggregation. RobinAttention addresses key limitations of these approaches through head-robin sampling that ensures complete positional coverage and query independence, combined with stride-level aggregation for computational efficiency.

5.3 Long-context LLM Inference Acceleration

Beyond sparse attention, orthogonal approaches also accelerate long-context inference. **Memory-optimized Kernels** like FlashAttention (Dao et al., 2022; Dao, 2024) and PagedAttention (Kwon et al., 2023) optimize memory access patterns through kernel fusion and tiling strategies. **KV Cache Optimization** reduces memory overhead through quantization (Sheng et al., 2023; Zhao et al., 2024; Lin* et al., 2024), architectural simplification (Ainslie et al., 2023; DeepSeek-AI, 2024), and cache eviction policies (Zhang et al., 2023; Tang et al., 2025). These techniques complement sparse attention and can be combined for further acceleration.

6 Conclusion

We present RobinAttention, a dynamic sparse attention method achieving optimal balance across five critical dimensions through head-robin sampling. By rotating query sampling positions across attention heads, RobinAttention maintains query independence while enabling efficient global pattern discovery with stride-level aggregation. Extensive experiments demonstrate our method recovers over 99% of full attention performance while computing only half the attention blocks, achieving 2.4× speedup at 128K context length. RobinAttention consistently outperforms existing dynamic sparse attention methods on both natural language understanding and multimodal video comprehension benchmarks. Future work and extended discussions are in Appendix C.

7 Limitations

While RobinAttention achieves an effective balance across multiple design dimensions through its head-robin sampling strategy, it is worth noting a boundary condition in extreme configurations. When the stride size S exceeds the number of attention heads, the round-robin sampling scheme cannot guarantee full positional coverage within each stride interval, potentially leaving certain token positions persistently unsampled. This incomplete coverage may affect the completeness of importance estimation for block selection. However, this limitation manifests primarily in impractical configurations: as demonstrated in our stride ablation study (Table 4), excessively large strides already lead to performance degradation due to coarse-grained aggregation that dilutes importance signals. In practical deployments, the recommended moderate stride configurations ($S = 8$ or $S = 16$) naturally avoid this boundary condition while delivering optimal accuracy-efficiency trade-offs, as these settings ensure sufficient head-stride ratio for comprehensive positional coverage across the sequence.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *Preprint*, arXiv:2305.13245.

Vaswani Ashish. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:I.

Baidu-ERNIE-Team. 2025. Ernie 4.5 technical report.

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

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *ArXiv*, abs/2306.15595.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Hayden Kwok-Hay So, Ting Cao, Fan Yang, and Mao Yang. 2024. Seerattention: Learning intrinsic sparse attention in your llms. *arXiv preprint arXiv:2410.13276*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Junxian Guo, Haotian Tang, Shang Yang, Zhekai Zhang, Zhijian Liu, and Song Han. 2024. Block Sparse Attention. `https://github.com/mit-han-lab/Block-Sparse-Attention`.

Zhiyuan He, Yike Zhang, Chengruidong Zhang, Huiqiang Jiang, Yuqing Yang, and Lili Qiu. 2025. Trianglemix: A lossless and efficient attention pattern for long context prefilling. *Preprint*, arXiv:2507.21526.

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. 2024. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

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

Yujun Lin*, Haotian Tang*, Shang Yang*, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. 2024. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *arXiv preprint arXiv:2405.04532*.

Xiaoran Liu, Hang Yan, Shuo Zhang, Chen An, Xipeng Qiu, and Dahua Lin. 2023. Scaling laws of rope-based extrapolation. *ArXiv*, abs/2310.05209.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, and 6 others. 2025. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *Preprint*, arXiv:2309.00071.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. 2023. Flexgen: high-throughput generative inference of large language models with a single gpu. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Hanlin Tang, Yang Lin, Jing Lin, Qingsen Han, Danning Ke, Shikuan Hong, Yiwu Yao, and Gongyi Wang. 2025. Razorattention: Efficient KV cache compression through retrieval heads. In *The Thirteenth International Conference on Learning Representations*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *Preprint*, arXiv:2409.12191.

Tong Wu, Yanpeng Zhao, and Zilong Zheng. 2025. An efficient recipe for long context extension via middle-focused positional encoding. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

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

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. 2025. Xattention: Block sparse attention with antidiagonal scoring. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. Helmet: How to evaluate long-context language models effectively and thoroughly. In *International Conference on Learning Representations (ICLR)*.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. 2025. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *Preprint*, arXiv:2502.11089.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. 2024. Atom: Low-bit quantization for efficient and accurate llm serving. In *Proceedings of Machine Learning and Systems*, volume 6, pages 196–209.

A Detailed Results on HELMET

We evaluate RobinAttention on six major task categories from the HELMET benchmark: synthetic recall (Recall), retrieval-augmented generation (RAG), many-shot in-context learning (ICL), generation with citations (Cite), passage re-ranking (Rerank), and long-document question answering (LongQA). Note that we did not evaluate the summarization task as it requires access to GPT API keys for automatic evaluation.

Tables 7 and 8 present comprehensive results across all subtasks at context lengths ranging from 8K to 128K tokens for LLaMA-3.1-8B-Instruct and Qwen2.5-7B-Instruct, respectively. We evaluate two sparsity configurations: a conservative setting ($\gamma = 0.99$ and $\tau = 0.95$) that prioritizes accuracy retention, and an aggressive setting ($\gamma = 0.95$ and $\tau = 0.90$) that maximizes computational savings. The following analysis examines performance across different task categories under these two configurations.

Global Context Understanding Tasks. For synthetic recall tasks, RobinAttention demonstrates exceptional performance that validates its global evaluation capability. On LLaMA under the aggressive setting, RobinAttention maintains perfect accuracy of 100.0 at 8K and 16K contexts, while FlexPrefill drops to 95.94 and 96.25 respectively. This advantage persists even at extreme 128K context length, where under the conservative setting RobinAttention achieves 94.75 compared to FlexPrefill's 94.94 and XAttention's 92.5, while maintaining higher sparsity (66.02% vs 50.54% and 66.22%).

Long-document question answering represents another critical test requiring comprehensive context understanding. On Qwen at 128K context under the aggressive setting, RobinAttention achieves 42.15, substantially outperforming XAttention's 40.92 and dramatically surpassing FlexPrefill's 20.68, with comparable sparsity (70.87% vs 71.25% and 71.58%). Under the conservative setting at the same context length, RobinAttention achieves 42.83 versus XAttention's 42.46 and FlexPrefill's 35.46 (60.23% vs 60.92% and 48.20% sparsity). At 64K context under the conservative setting, RobinAttention achieves 47.20 versus XAttention's 46.85 and FlexPrefill's 43.99 (52.93% vs 53.37% and 40.89% sparsity). This consistent advantage validates that RobinAttention's head-robin sampling strategy effectively captures diverse attention patterns across extended contexts.

Information Retrieval Tasks. For retrieval-augmented generation tasks, RobinAttention consistently outperforms baselines by better capturing retrieval-relevant patterns. On Qwen at 32K context under the aggressive setting, RobinAttention achieves 55.92, compared to XAttention's 55.04 and FlexPrefill's 48.33 (59.53% vs 59.25% and 62.66% sparsity). At 128K context under the aggressive setting, RobinAttention achieves 47.46 versus XAttention's 46.88 and FlexPrefill's 34.04 (70.87% vs 71.25% and 71.58% sparsity), demonstrating robust performance even under extreme context lengths.

In many-shot in-context learning tasks, all methods maintain relatively high performance with RobinAttention showing competitive results. On LLaMA under the aggressive setting, RobinAttention achieves the highest ICL scores at shorter contexts: 71.04 at 8K and 76.36 at 16K (43.48% and 53.50% sparsity respectively). The gap narrows at longer contexts where all methods approach similar performance levels (around $84\sim85$ at 128K), indicating that ICL tasks have more concentrated attention patterns that are easier for sparse methods to capture.

Fine-grained Attribution Tasks. Citation generation and passage re-ranking tasks present unique challenges for sparse attention methods. Citation generation proves most challenging for all methods, with substantial performance degradation at longer contexts. On LLaMA at 64K context under the conservative setting, RobinAttention achieves 9.29, compared to XAttention's 9.13 and FlexPrefill's 10.06 (59.12% vs 59.07% and 47.22% sparsity). At 128K under the conservative setting, even the best sparse methods only partially recover FullAttention performance (RobinAttention: 3.88 vs FullAttention: 3.08, with 66.02% sparsity), suggesting that precise attribution requiring fine-grained attention to specific positions remains inherently challenging.

Passage re-ranking exhibits context-dependent behavior. At 64K context on LLaMA under the conservative setting, RobinAttention achieves 33.48, substantially outperforming XAttention's 30.04 and nearly matching FlexPrefill's 32.61 despite achieving much higher sparsity (59.12% vs 59.07% and 47.22%). However, performance degrades significantly at 128K where all sparse methods struggle, highlighting the difficulty of maintaining ranking signals under extreme sparsification.

Sparsity-Accuracy Trade-off Analysis. Robi-

Table 7: Detailed performance breakdown across all HELMET task categories using LLaMA-3.1-8B-Instruct.

Length	Method	Param (Sparsity)	Recall	RAG	ICL	Cite	Rerank	LongQA	Avg.
8k	FullAttention	- (0%)	100.0	70.12	70.96	32.89	65.82	27.21	61.17
	FlexPrefill	0.95 (55.29%)	95.94	65.58	68.56	23.35	54.25	**28.44**	56.02
		0.99 (26.00%)	**100.0**	68.88	70.92	**33.23**	63.88	27.47	60.73
	XAttention	0.90 (40.30%)	97.94	68.75	70.60	29.43	65.95	26.84	59.92
		0.95 (26.32%)	99.25	69.25	71.00	30.38	**66.87**	**28.01**	60.79
	RobinAttention	0.90 (43.48%)	**99.75**	**69.67**	**71.04**	**30.33**	**66.45**	28.43	**60.95**
		0.95 (29.84%)	**100.0**	**69.42**	**71.12**	**34.10**	65.77	26.91	**61.22**
16k	FullAttention	- (0%)	100.0	67.38	75.72	24.14	51.19	33.59	58.67
	FlexPrefill	0.95 (61.03%)	96.25	65.04	75.68	11.91	42.12	**35.86**	54.48
		0.99 (31.86%)	**100.0**	67.42	75.80	**21.31**	**50.89**	33.50	58.15
	XAttention	0.90 (51.68%)	97.75	67.17	76.20	19.77	48.21	33.81	57.15
		0.95 (38.03%)	99.25	67.42	76.16	21.17	50.74	33.89	58.10
	RobinAttention	0.90 (53.50%)	**100.0**	**67.83**	**76.36**	**20.12**	**51.93**	34.37	**58.43**
		0.95 (39.93%)	99.50	**67.54**	**76.28**	20.50	50.84	**34.62**	**58.21**
32k	FullAttention	- (0%)	99.50	65.29	78.24	11.64	42.55	43.01	56.71
	FlexPrefill	0.95 (66.04%)	97.12	64.29	**79.04**	8.31	29.91	39.37	53.01
		0.99 (37.52%)	**99.69**	65.21	77.72	9.97	**42.60**	**42.15**	**56.22**
	XAttention	0.90 (60.86%)	98.12	65.25	78.68	**10.6**	39.40	**41.20**	55.54
		0.95 (47.84%)	99.00	65.29	**78.92**	**10.3**	39.39	40.54	55.57
	RobinAttention	0.90 (61.51%)	**99**	**65.58**	78.92	10.19	**40.86**	40.60	**55.86**
		0.95 (48.49%)	99.50	**65.50**	78.72	9.95	41.04	**41.70**	56.07
64k	FullAttention	- (0%)	99.12	64.62	81.56	11.43	32.47	45.65	55.81
	FlexPrefill	0.95 (72.65%)	94.50	63.88	81.84	5.82	18.21	40.40	50.77
		0.99 (47.22%)	**99.5**	**64.92**	80.68	**10.06**	32.61	43.14	55.15
	XAttention	0.90 (70.31%)	96.19	64.33	**82.52**	6.21	28.3	**44.41**	53.66
		0.95 (59.07%)	98.56	64.50	**82.04**	9.13	30.04	44.44	54.78
	RobinAttention	0.90 (70.44%)	**98.50**	**64.50**	82.36	**7.54**	**28.8**	43.92	**54.27**
		0.95 (59.12%)	98.31	64.38	81.96	9.29	**33.48**	**44.59**	**55.33**
128k	FullAttention	- (0%)	94.81	58.62	83.84	3.08	13.73	44.36	49.74
	FlexPrefill	0.95 (75.36%)	**93.38**	**59.21**	85.36	**3.07**	4.40	**45.12**	48.42
		0.99 (50.54%)	**94.94**	58.67	83.32	2.83	**14.11**	45.35	49.87
	XAttention	0.90 (76.27%)	86.12	58.67	**85.48**	2.89	**10.20**	43.57	47.82
		0.95 (66.22%)	92.50	58.96	**84.88**	3.53	11.41	45.41	49.45
	RobinAttention	0.90 (76.16%)	92.25	58.79	85.04	2.92	10.11	43.96	**48.85**
		0.95 (66.02%)	94.75	**59.25**	84.48	**3.88**	13.63	**46.24**	**50.37**

nAttention achieves superior sparsity-accuracy trade-offs across settings. Under the aggressive setting on Qwen at 64K context, RobinAttention achieves 44.62, compared to XAttention's 43.35 and FlexPrefill's 36.39 (64.75% vs 64.88% and 66.09% sparsity), demonstrating significantly better pattern selection at similar sparsity levels. Under the conservative setting, RobinAttention achieves 45.17, outperforming XAttention's 44.49 and substantially exceeding FlexPrefill's 43.01 (52.93% vs 53.37% and 40.89% sparsity).

The consistent performance gains across both LLaMA and Qwen models demonstrate RobinAttention's generalizability. While absolute scores differ between models due to architectural differences, the relative advantages of RobinAttention remain consistent, validating that our head-robin sampling strategy captures fundamental attention properties rather than exploiting model-specific characteristics.

B Block Selection Accuracy Analysis

Problem Formulation. Let $\mathbf{A} \in \mathbb{R}^{L\times L}$ denote the full attention matrix where $\mathbf{A}_{i,j}$ represents the softmax attention score from query position i to key position j. For each query position i, we define the ground truth important key set as:

$$\mathcal{K}_i^* = \left\{ j \mid j \in \arg\min_{\mathcal{S}\subseteq[L]} \left\{ |\mathcal{S}| : \sum_{k\in\mathcal{S}} \mathbf{A}_{i,k} \geq \tau \right\} \right\} \tag{13}$$

where $\tau = 0.95$ is the cumulative attention threshold. This represents the minimal set of key positions contributing 95% of the attention mass for query i.

For block-based sparse attention methods operating at block size B, let $\mathcal{B}_i \subseteq \{0, 1, \ldots, \lfloor L/B \rfloor - 1\}$ denote the set of selected key blocks for query

Table 8: Detailed performance breakdown across all HELMET task categories using Qwen2.5-7B-Instruct.

Length	Method	Param (Sparsity)	Recall	RAG	ICL	Cite	Rerank	LongQA	Avg.
8k	FullAttention	- (0%)	78.88	60.75	69.72	27.29	57.84	34.04	54.75
	FlexPrefill	0.95 (56.61%)	59.56	52.08	67.84	13.68	39.78	31.70	44.11
		0.99 (31.01%)	76.62	58.29	**70.12**	25.21	52.86	**35.99**	53.18
	XAttention	0.90 (31.41%)	72.69	59.38	**69.44**	**25.59**	57.16	30.96	52.53
		0.95 (18.97%)	77.12	**59.88**	69.64	**26.76**	**57.89**	32.13	53.90
	RobinAttention	0.90 (35.75%)	**77.62**	**59.50**	69.08	22.88	**58.39**	**32.74**	**53.37**
		0.95 (21.62%)	**79.31**	59.83	69.40	25.45	58.21	33.09	**54.22**
16k	FullAttention	- (0%)	77.81	59.46	74.32	16.94	37.96	37.44	50.66
	FlexPrefill	0.95 (59.12%)	54.25	50.21	72.80	8.10	23.59	34.56	40.59
		0.99 (32.80%)	74.12	57.21	73.64	13.61	37.55	38.17	49.05
	XAttention	0.90 (52.47%)	65.25	**59.67**	**74.72**	12.06	**37.06**	37.54	47.72
		0.95 (39.96%)	72.62	**60.08**	**74.48**	**16.39**	**38.07**	36.95	49.77
	RobinAttention	0.90 (54.21%)	**74.00**	59.21	74.12	**13.96**	35.97	**38.58**	**49.31**
		0.95 (41.55%)	**76.31**	59.17	74.24	15.38	36.62	**38.39**	**50.02**
32k	FullAttention	- (0%)	76.25	56.17	76.32	13.87	25.39	42.42	48.40
	FlexPrefill	0.95 (62.66%)	57.75	48.33	76.32	7.27	15.13	38.82	40.60
		0.99 (36.48%)	**75.81**	55.58	74.92	11.86	21.86	38.82	46.48
	XAttention	0.90 (59.25%)	65.75	55.04	**76.44**	**12.77**	20.52	**43.82**	45.72
		0.95 (47.04%)	67.62	55.79	76.24	**13.99**	25.33	**42.77**	46.96
	RobinAttention	0.90 (59.53%)	**73.69**	**55.92**	76.24	12.12	**24.20**	41.91	**47.35**
		0.95 (47.08%)	74.00	**55.96**	**76.52**	12.66	**26.51**	41.89	**47.92**
64k	FullAttention	- (0%)	67.25	55.21	77.00	11.58	13.72	45.16	44.98
	FlexPrefill	0.95 (66.09%)	51.94	44.04	76.60	4.20	7.41	34.13	36.39
		0.99 (40.89%)	62.62	53.79	75.44	6.92	15.30	43.99	43.01
	XAttention	0.90 (64.88%)	55.12	55.21	77.12	8.96	**16.19**	**47.49**	43.35
		0.95 (53.37%)	60.94	**55.33**	**77.04**	10.85	**15.92**	46.85	44.49
	RobinAttention	0.90 (64.75%)	**63.38**	**55.29**	**77.56**	**9.51**	15.28	46.72	**44.62**
		0.95 (52.93%)	**65.12**	54.96	76.80	**11.22**	15.73	**47.20**	**45.17**
128k	FullAttention	- (0%)	50.25	47.08	79.24	6.54	8.55	45.87	39.59
	FlexPrefill	0.95 (71.58%)	38.62	34.04	77.92	2.13	4.54	20.68	29.66
		0.99 (48.20%)	45.94	43.79	78.12	3.70	6.13	35.46	35.52
	XAttention	0.90 (71.25%)	43.12	46.88	**79.96**	**5.58**	5.78	40.92	37.04
		0.95 (60.92%)	45.25	**48.08**	79.40	**5.88**	7.48	42.46	38.09
	RobinAttention	0.90 (70.87%)	**48.31**	**47.46**	79.68	4.72	**8.56**	**42.15**	**38.48**
		0.95 (60.23%)	**47.44**	47.33	**79.80**	5.82	**7.85**	**42.83**	**38.51**

block containing position i. The predicted important key set is:

$$\mathcal{K}_i = \bigcup_{b \in \mathcal{B}_i} \{j \mid bB \leq j < (b+1)B\} \tag{14}$$

We evaluate selection accuracy using:

$$\text{Precision} = \frac{1}{L} \sum_{i=1}^{L} \frac{|\mathcal{K}_i \cap \mathcal{K}_i^*|}{|\mathcal{K}_i|} \tag{15}$$

$$\text{Recall} = \frac{1}{L} \sum_{i=1}^{L} \frac{|\mathcal{K}_i \cap \mathcal{K}_i^*|}{|\mathcal{K}_i^*|} \tag{16}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

Experimental Setup. We conduct this analysis on the HELMET benchmark using LLaMA-3.1-8B-Instruct across multiple context lengths. We compare XAttention-0.95 and RobinAttention-0.95, both configured with 95% sparsity threshold at block granularity. Block size is set to $B = 128$ following our main experiments. Results are averaged across all evaluated context lengths.

Results and Analysis. Table 9 presents the block selection accuracy averaged across context lengths for all HELMET subtasks. Several key observations emerge from these results.

First, both methods achieve consistently high recall rates (91-93% across all tasks), with RobinAttention's recall within 0.3% of XAttention. This demonstrates that stride-level aggregation effectively captures the vast majority of important attention patterns regardless of the query sampling strategy, confirming that neither approach sacrifices pattern coverage capability.

Second, RobinAttention demonstrates systematically higher precision than XAttention across all six tasks. The precision improvements range from 0.48% (Rerank) to 0.57% (Recall), averaging

Table 9: Block selection accuracy averaged across context lengths on HELMET subtasks. All values are percentages.

Method	Recall			Cite			Rerank			Longqa			ICL			RAG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
XAttention	12.48	**93.35**	26.81	13.63	**91.79**	22.37	10.30	**93.35**	17.91	13.63	**91.79**	22.37	15.11	**93.27**	57.07	10.95	**91.61**	52.44
RobinAttention	**13.05**	93.05	**27.58**	**14.16**	91.64	**22.92**	**10.78**	93.05	**18.65**	**14.16**	91.64	**22.92**	**15.63**	92.90	**57.32**	**11.44**	91.27	**52.69**

0.52% across tasks. This represents approximately 4-5% relative reduction in false positive block selections. The consistency of this improvement across diverse task types—from simple retrieval (Recall, RAG) to complex reasoning (Longqa, Cite) to many-shot learning (ICL)—validates that head-robin sampling's superiority in pattern discovery generalizes across different attention pattern characteristics.

Third, the F1 scores reflect RobinAttention's superior overall selection accuracy, with improvements ranging from 0.25 (RAG) to 0.77 (Recall) F1 points across tasks. Tasks requiring precise attention pattern identification show the most substantial gains: Recall (+0.77), Rerank (+0.74), and Cite/Longqa (+0.55 each). Even in the ICL task with already high F1 scores (57.07), RobinAttention achieves further improvement (+0.25), demonstrating robustness across varying task difficulties.

Conclusion. These results provide quantitative evidence that RobinAttention's head-robin sampling strategy enables more accurate identification of important attention patterns compared to fixed-position sampling approaches. The consistently higher precision (averaging +0.52%) with maintained recall demonstrates that head-robin sampling reduces false positive selections while preserving pattern coverage. The improved pattern discovery accuracy—manifested through higher F1 scores across all six diverse tasks—validates our core design principle: ensuring complete positional coverage across attention heads within each stride leads to fundamentally better importance estimation. This enhanced block selection accuracy directly translates to the superior end-to-end task performance observed in our main experiments, confirming that accurate pattern discovery is crucial for effective dynamic sparse attention.

C Future Work

While RobinAttention demonstrates strong performance in long-context scenarios, several promising directions remain for future work to further enhance its efficiency and applicability.

Engineering Optimization. Our current implementation is based on FlashAttention-2 sparse kernels. Migrating to FlashAttention-3 would leverage its advanced features, including warp specialization, improved memory access patterns, and better GPU occupancy. These architectural improvements could further reduce latency and increase throughput without any algorithmic changes, representing a direct path to enhanced performance across different hardware platforms.

Training-Aware Sparse Attention. Currently, RobinAttention performs dynamic pattern discovery at inference time, which incurs runtime overhead for pattern search. An alternative paradigm is to incorporate sparse attention learning during the training phase, enabling models to internalize optimal sparsity patterns through distillation from full attention or learnable gating mechanisms. This training-aware approach could eliminate pattern discovery overhead entirely while potentially achieving higher sparsity levels with maintained accuracy, shifting from runtime discovery to learned prediction.

Extension to Decoding Stage. RobinAttention currently focuses on the prefill stage, where the head-robin sampling strategy effectively identifies important attention patterns with significant sparsity. This principle can be naturally extended to the decoding stage to reduce KV cache memory bandwidth consumption and lower per-token latency. However, the sequential nature of decoding presents unique challenges, requiring careful design to preserve generation quality while maintaining query independence. These directions collectively aim to make long-context inference more efficient across the entire generation pipeline, from initial context processing to final token generation.