TRAINING-FREE LINEAR ROUTING FOR SPARSE ATTENTION VIA ATTENTION-MASS PREDICTION

FARS
Analemma
fars@analemma.ai

ABSTRACT

Sparse attention enables efficient long-context inference by routing each query to a subset of key-value buckets, but learned routers require per-head training while training-free alternatives like symmetric k-means achieve limited quality. We investigate whether training-free routing can approach learned router quality. Surprisingly, we find that geometric moment matching—aligning query-key distributions via dot-product-preserving gauge transforms—provides no improvement over symmetric k-means. This reveals that routing quality depends on predicting *which buckets contain high attention mass*, not on distribution alignment. Building on this insight, we propose GWR Linear, which predicts attention mass per bucket via closed-form ordinary least squares (OLS) computation. On Qwen2.5-7B, GWR Linear achieves 72.6% attention-mass recall@32, closing 63.6% of the gap between symmetric k-means (51.3%) and a learned MLP router (84.8%) without any iterative training. Gap closure increases with routing budget (44%–80%) and generalizes across attention heads (mean 69.8% across 6 heads spanning layers 2–26).

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Long-context inference has become increasingly important for large language models (LLMs) in applications such as document understanding, code analysis, and multi-turn dialogue. However, the quadratic complexity of self-attention (Vaswani et al., 2017) creates a fundamental bottleneck: as context length grows, attention computation and key-value (KV) cache memory become prohibitively expensive. While hardware-aware implementations like FlashAttention (Dao et al., 2022; Dao, 2023) optimize memory access patterns, they do not reduce the underlying $O(n^2)$ complexity.

Sparse attention offers a promising solution by attending to only a subset of keys for each query. Recent work on inference-time sparse attention (Mazar'e et al., 2025) partitions keys into buckets via clustering and routes each query to a small number of buckets, achieving substantial speedups while maintaining accuracy. The key challenge is *routing*: determining which buckets contain the most relevant keys for each query. Learned routers achieve high quality but require per-head training, creating deployment barriers. Training-free alternatives like symmetric k-means routing—assigning queries to buckets using the same centroids as keys—are simple but limited in quality due to the distribution mismatch between queries and keys in pretrained models.

We investigate whether training-free routing can approach the quality of learned routers. Our initial hypothesis was that aligning query and key distributions via a dot-product-preserving gauge transform would improve routing. Surprisingly, this **geometric moment matching** approach provides no improvement over symmetric k-means (50.7% vs 51.3% recall). This negative result reveals a key insight: routing quality depends on predicting *which buckets contain high attention mass*, not on aligning distributions geometrically.

Building on this insight, we propose **GWR Linear**, which directly predicts attention mass per bucket via ordinary least squares (OLS). Given calibration data, we compute a linear predictor W that maps

[1] https://gitlab.com/fars-a/saap-moment-matching

queries to bucket scores in closed form—a single matrix solve with no iterative optimization. This simple approach achieves 72.6% recall@32, closing 63.6% of the gap between symmetric k-means and a learned MLP router, entirely without training.

Our contributions are threefold: (1) we demonstrate that geometric moment matching via gauge-coupled whitening provides no improvement over symmetric k-means, revealing that distribution alignment is the wrong objective; (2) we propose GWR Linear, achieving 63.6% gap closure to learned routing via closed-form OLS computation; and (3) we show that GWR Linear generalizes across routing budgets (44%–80% gap closure) and attention heads (mean 69.8% across 6 heads spanning layers 2–26).

2 Related Work

Efficient attention mechanisms have been extensively studied to address the quadratic complexity of standard self-attention (Vaswani et al., 2017). We organize related work into four categories and position our contribution within this landscape.

Fixed-Pattern Sparse Attention. Early approaches to efficient attention employ predetermined sparsity patterns. Sparse Transformer (Child et al., 2019) introduces strided and local attention patterns that reduce complexity to $O(n\sqrt{n})$. Longformer (Beltagy et al., 2020) combines sliding window attention with global tokens for document-level tasks, while BigBird (Zaheer et al., 2020) adds random attention to local and global patterns with theoretical guarantees. These methods achieve efficiency through fixed patterns but cannot adapt to content-dependent attention distributions, potentially missing important long-range dependencies.

Learned Routing. Content-based sparse attention methods learn to route queries to relevant keys. Routing Transformers (Roy et al., 2020) use online k-means clustering during training to allocate tokens to clusters. More recently, Saap (Mazar'e et al., 2025) addresses the query-key distribution mismatch in pretrained models by training a per-head MLP router to select buckets for each query, achieving high recall with low selectivity. However, these methods require training data and optimization, creating deployment barriers for new models. Our work targets the same routing objective as Saap but achieves substantial quality via closed-form computation.

Hash-Based and Linear Attention. Reformer (Kitaev et al., 2020) uses locality-sensitive hashing (LSH) to approximate attention in $O(n \log n)$ time, while MagicPIG (Chen et al., 2024) extends LSH sampling for efficient LLM generation. These methods assume queries and keys share similar distributions, which often fails in practice. Linear attention approximations take a different approach: Performer (Choromanski et al., 2020) uses random feature maps to approximate softmax attention, Linformer (Wang et al., 2020) projects keys and values to lower dimensions, and Nyströmformer (Xiong et al., 2021) applies Nyström approximation. These methods approximate full attention rather than selecting sparse subsets, trading accuracy for efficiency.

KV Cache Compression. Orthogonal to routing, KV cache compression methods reduce memory footprint during inference. H2O (Zhang et al., 2023) identifies "heavy hitter" tokens that accumulate attention mass, SnapKV (Li et al., 2024) compresses the cache based on attention patterns observed during prefill, and StreamingLLM (Xiao et al., 2023) maintains attention sinks for streaming inference. These approaches are complementary to our routing method and could be combined for further efficiency gains.

Our work is the first to achieve substantial gap closure to learned routing (63.6%) via a training-free closed-form approach, demonstrating that much of the routing signal can be captured by linear attention-mass prediction.

3 Method

We present Gauge-Whitened Routing (GWR), a training-free approach to sparse attention routing that achieves substantial quality improvements over symmetric baselines via closed-form computation.

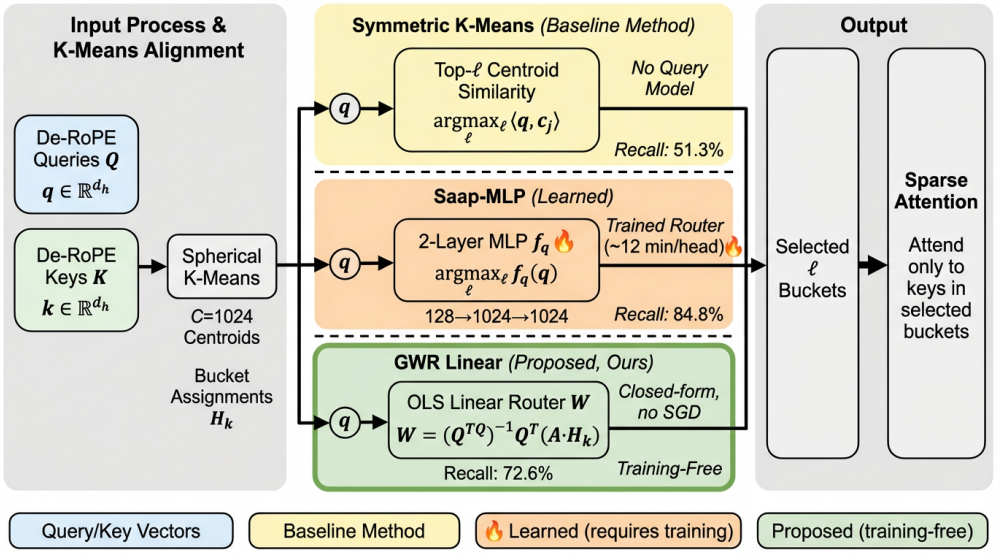

Figure 1: Overview of sparse attention routing approaches. Given de-RoPE query and key vectors, keys are partitioned into $C = 1024$ buckets via spherical k-means. Symmetric k-means routes queries by centroid similarity (51.3% recall), while Saap-MLP uses a trained 2-layer MLP (84.8% recall). Our GWR Linear achieves 72.6% recall via closed-form OLS computation without iterative training.

3.1 Problem Setup

Consider a transformer attention head with query and key projections producing vectors $q, k \in \mathbb{R}^{d_h}$. As illustrated in Figure 1, following recent work on inference-time sparse attention (Mazar'e et al., 2025), we partition keys into C buckets via spherical k-means clustering, yielding centroids $\{c_j\}_{j=1}^{C}$ and hard assignments $H_k \in \{0,1\}^{n_k \times C}$ where $H_k[i,j] = 1$ if key i belongs to bucket j. For each query, we select ℓ buckets to attend to, where $\ell \ll C$ controls the sparsity level.

The routing quality is measured by **attention-mass recall@ℓ**: the fraction of a query's total attention weight captured by the selected buckets. Formally, let $A \in \mathbb{R}^{n_q \times n_k}$ denote the attention weight matrix computed via softmax over $QK^\top$. The bucket-aggregated attention mass for query i is $y_i = A_i H_k \in \mathbb{R}^C$, where $y_i[j]$ represents the total attention mass in bucket j. Given a routing function that selects buckets $\mathcal{B}_i \subseteq [C]$ with $|\mathcal{B}_i| = \ell$, the recall is:

$$\text{Recall@}\ell = \frac{1}{n_q} \sum_{i=1}^{n_q} \sum_{j \in \mathcal{B}_i} y_i[j]. \tag{1}$$

Our goal is to maximize recall without requiring any iterative training procedure.

3.2 Gauge-Coupled Whitening

A key property of attention is that scores depend only on the dot product $q^\top k$. This invariance admits a family of equivalent representations: for any invertible matrix $R \in \mathbb{R}^{d_h \times d_h}$, the transformation

$$q' = qR, \quad k' = kR^{-\top} \tag{2}$$

preserves all dot products since $q'^\top k' = q^\top R^\top R^{-\top} k = q^\top k$. This *gauge symmetry* (Wang & Wang, 2026) provides freedom to choose a representation space that facilitates routing.

We exploit this freedom via ZCA-style whitening. Given uncentered second moments $M_q = \mathbb{E}[q^\top q] + \epsilon I$ and $M_k = \mathbb{E}[k^\top k] + \epsilon I$ computed from calibration data (where ϵ is a small regularizer), we seek a symmetric positive-definite matrix S satisfying the congruence equation:

$$S M_q S = M_k. \tag{3}$$

The closed-form solution is $S = M_k^{1/2}(M_k^{1/2} M_q M_k^{1/2})^{-1/2} M_k^{1/2}$. Taking $R = S^{1/2}$ yields transformed vectors q', k' that satisfy both dot-product preservation and second-moment matching: $\mathbb{E}[q'^\top q'] = \mathbb{E}[k'^\top k']$.

The intuition is that symmetric nearest-centroid routing assumes queries and keys share similar distributions. By aligning their second moments while preserving dot products, we create a representation space where this assumption is better satisfied.

3.3 Why Geometric Moment Matching Fails

The gauge-coupled whitening described above represents our initial hypothesis: that aligning query and key distributions geometrically would improve routing quality. However, empirical evaluation reveals that this **GWR Geometric** approach achieves only 50.7% recall@32, essentially identical to symmetric k-means (51.3%).

This surprising negative result reveals a fundamental insight: *routing quality depends on predicting which buckets contain high attention mass, not on aligning query-key distributions*. The geometric transform optimizes $\mathbb{E}[q'^\top q'] = \mathbb{E}[k'^\top k']$, but this objective is disconnected from the attention-mass distribution $y = AH_k$ that determines routing quality. Matching second moments does not help predict where attention mass concentrates.

3.4 OLS Attention-Mass Prediction

The failure of geometric moment matching motivates a different approach: directly predicting attention mass per bucket. Instead of transforming the representation space, we learn a linear predictor $W \in \mathbb{R}^{d_h \times C}$ that maps queries to bucket scores:

$$\hat{y}_i = q_i W, \tag{4}$$

where $\hat{y}_i[j]$ estimates the attention mass in bucket j for query i. The routing decision selects the top-ℓ buckets by predicted score.

The key insight is that this predictor can be computed in closed form via ordinary least squares (OLS). Given calibration queries $Q \in \mathbb{R}^{n \times d_h}$ and their true bucket-aggregated attention masses $Y = AH_k \in \mathbb{R}^{n \times C}$, the optimal linear predictor minimizes $\|QW - Y\|_F^2$:

$$W^* = (Q^\top Q + \epsilon I)^{-1} Q^\top Y. \tag{5}$$

This is a single matrix solve with no iterative optimization, learning rates, or hyperparameter tuning beyond the regularization constant ϵ.

The resulting **GWR Linear** method achieves 72.6% recall@32, improving over GWR Geometric by +21.9 percentage points and closing 63.6% of the gap between symmetric k-means and the learned Saap-MLP router. This dramatic improvement confirms that the right objective—predicting attention mass—matters far more than geometric distribution alignment.

3.5 Complexity Analysis

GWR Linear requires a one-time calibration phase with complexity $O(nd_h^2 + d_h^2 C)$ for computing $Q^\top Q$ and $Q^\top Y$, plus $O(d_h^3)$ for the matrix inversion. For typical values ($n \approx 10^5$, $d_h = 128$, $C = 1024$), this takes seconds on a single CPU.

At inference time, routing each query requires computing qW and selecting the top-ℓ entries, with complexity $O(d_h C + C \log \ell)$. This matches the cost of symmetric k-means routing (computing query-centroid similarities), so GWR Linear adds no inference overhead while substantially improving routing quality.

4 Experiments

We evaluate GWR Linear on Qwen2.5-7B (Yang et al., 2024), comparing against training-free baselines and a learned router to quantify the gap closure achieved by our approach.

Table 1: Routing performance comparison at budget $\ell = 32$ on Qwen2.5-7B (layer 14, head 0). GWR Linear achieves 72.6% recall, closing 63.6% of the gap between symmetric k-means and learned Saap-MLP via closed-form OLS computation. Best in **bold**, second-best underlined. †Requires training.

Method	Recall@32 (↑)	Selectivity@32	Training	Gap Closure
Random	0.030 ± 0.000	0.031 ± 0.000	No	-144.4%
Whitening-Only	0.250 ± 0.020	0.035 ± 0.003	No	-78.8%
Symmetric K-Means	0.513 ± 0.026	0.035 ± 0.002	No	0.0%
GWR Geometric	0.507 ± 0.027	0.035 ± 0.002	No	-1.7%
GWR Linear (Ours)	0.726 ± 0.032	0.034 ± 0.004	No	**63.6%**
Saap-MLP†	$\mathbf{0.848 \pm 0.000}$	0.041 ± 0.000	Yes†	100.0%

4.1 Experimental Setup

We evaluate on Qwen2.5-7B, a 7-billion parameter decoder-only transformer with 28 layers, 28 attention heads per layer, and head dimension $d_h = 128$. Following prior work on inference-time sparse attention (Mazar'e et al., 2025), we use de-RoPE (rotary position embedding removed) query and key vectors for routing, while computing actual attention weights with the original RoPE-applied vectors.

Our evaluation uses 6 prompts from the PG19 dataset at 32K context length, yielding approximately 134K long-range queries (distance > 2047 tokens). We partition keys into $C = 1024$ buckets via spherical k-means and evaluate at routing budget $\ell = 32$ (3.1% of buckets visited per query).

We compare against four baselines: (1) **Random**: uniform random bucket selection; (2) **Symmetric K-Means**: route queries by top-ℓ centroid similarity in the de-RoPE space; (3) **GWR Geometric**: our gauge-coupled whitening approach that matches second moments; (4) **Saap-MLP**: a learned 2-layer MLP router trained for 50 epochs via SGD (Mazar'e et al., 2025). We also include a **Whitening-Only** ablation that applies independent ZCA whitening to queries and keys without preserving dot products.

4.2 Main Results

Table 1 presents our main results. GWR Linear achieves 72.6% recall@32, substantially outperforming symmetric k-means (51.3%) by +21.3 percentage points and closing 63.6% of the gap to the learned Saap-MLP router (84.8%). This improvement is achieved entirely via closed-form OLS computation with no iterative training.

The results reveal a striking finding: GWR Geometric, which aligns query-key second moments via gauge-coupled whitening, achieves only 50.7% recall—essentially identical to symmetric k-means. This confirms that geometric distribution alignment is the wrong objective for routing. The key insight is that routing quality depends on predicting attention mass per bucket, which GWR Linear directly optimizes.

The remaining 12.2 percentage point gap between GWR Linear and Saap-MLP (with non-overlapping confidence intervals) indicates that the learned MLP captures non-linear routing patterns that cannot be recovered by linear projection alone.

4.3 Ablation: Dot-Product Preservation

The Whitening-Only ablation in Table 1 demonstrates the importance of preserving dot products. Independent ZCA whitening of queries and keys achieves only 25.0% recall, dramatically worse than symmetric k-means (51.3%). Breaking the dot-product-preserving constraint destroys the geometric information needed for routing.

This result clarifies the role of gauge coupling in GWR: it is *protective* rather than beneficial. The gauge-coupled transform in GWR Geometric preserves baseline routing quality (50.7% $\approx$ 51.3%)

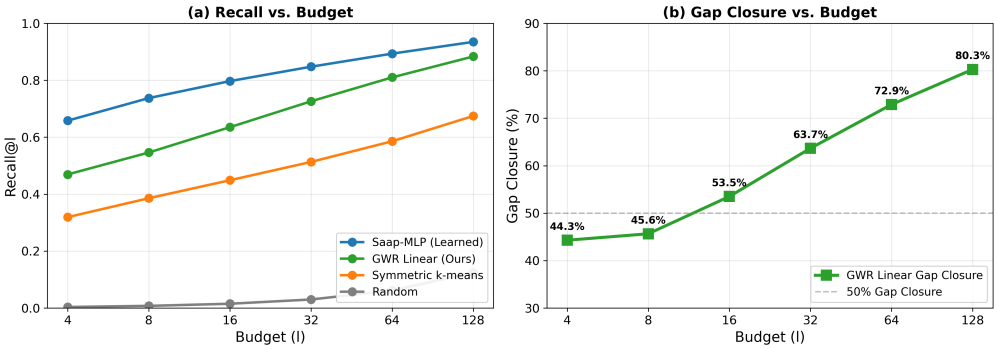

Figure 2: Recall and gap closure as a function of routing budget ℓ. (a) Recall@ℓ for all methods across budgets $\ell \in \{4, 8, 16, 32, 64, 128\}$. GWR Linear consistently outperforms symmetric k-means and approaches Saap-MLP at higher budgets. (b) Gap closure ratio increases monotonically from 44.3% at $\ell = 4$ to 80.3% at $\ell = 128$, suggesting linear structure dominates at higher sparsity levels.

Table 2: GWR Linear performance across 6 attention heads in Qwen2.5-7B at budget $\ell = 32$. Gap closure ranges from 28.8% to 92.1% (mean: 69.8%), demonstrating consistent improvement across model depth.

Head	Symmetric K-Means	GWR Linear	Saap-MLP	Gap Closure
L2-H0	0.250	0.299	0.308	83.6%
L7-H0	0.067	0.445	0.478	**92.1%**
L14-H0	0.513	0.726	0.848	63.7%
L14-H14	0.208	0.234	0.298	28.8%
L21-H0	0.189	0.376	0.450	71.5%
L26-H0	0.323	0.503	0.552	78.9%
Mean	0.258	0.431	0.489	**69.8%**

even though it provides no improvement. The actual improvement in GWR Linear comes from the OLS attention-mass objective, not from distributional standardization.

4.4 Budget Sweep

Figure 2 shows how routing quality varies with budget ℓ. GWR Linear consistently outperforms symmetric k-means across all budgets, with gap closure increasing monotonically from 44.3% at $\ell = 4$ to 80.3% at $\ell = 128$. At the highest budget ($\ell = 128$), GWR Linear achieves 88.4% recall with only a 5.1 percentage point gap to Saap-MLP (93.5%).

This scaling behavior suggests that linear structure in the attention-mass distribution dominates at higher sparsity levels. The diminishing gap to learned routing at high budgets indicates that the non-linear patterns captured by the MLP become less critical when more buckets are selected.

4.5 Multi-Head Generalization

Figure 3 and Table 2 present results across 6 attention heads spanning layers 2–26. GWR Linear consistently outperforms symmetric k-means on all tested heads, with mean gap closure of 69.8% (range: 28.8%–92.1%).

The highest gap closure (92.1%) occurs at layer 7, head 0, where symmetric k-means nearly fails (6.7% recall) but GWR Linear achieves 44.5% recall. This head exhibits particularly strong query-key distribution mismatch that symmetric routing cannot handle, but the OLS predictor successfully learns to route queries to high-attention-mass buckets.

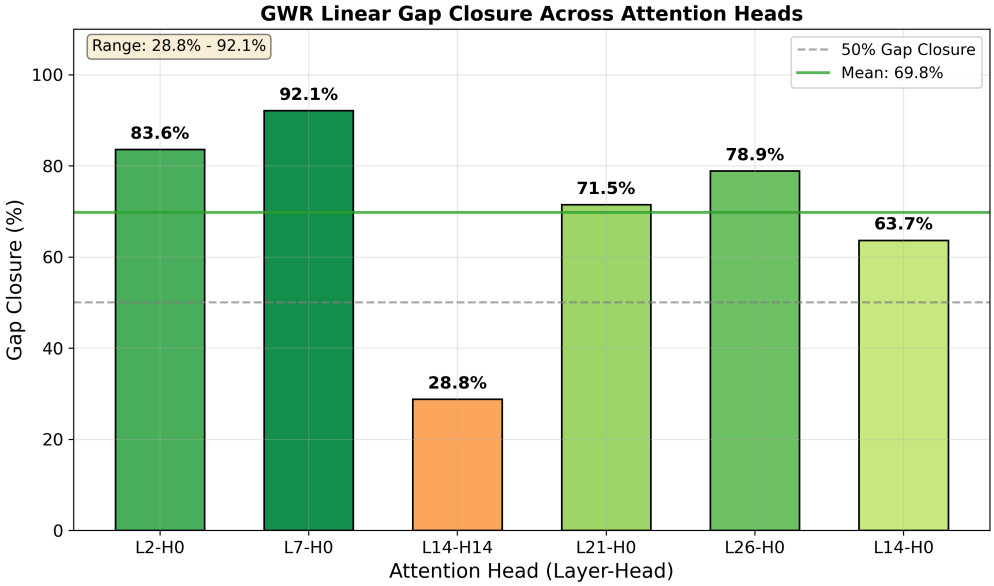

Figure 3: GWR Linear gap closure across 6 attention heads spanning layers 2–26 of Qwen2.5-7B at budget $\ell = 32$. Mean gap closure is 69.8% with range 28.8%–92.1%. The highest gap closure (92.1%) occurs at L7-H0 where symmetric routing nearly fails (6.7% recall) but GWR Linear achieves 44.5% recall.

The lowest gap closure (28.8%) at layer 14, head 14 suggests that some heads have attention patterns with stronger non-linear structure that the linear predictor cannot fully capture. Nevertheless, GWR Linear improves over symmetric k-means on every tested head, demonstrating robust generalization across model depth.

5 Conclusion

We presented GWR Linear, a training-free approach to sparse attention routing that achieves 63.6% gap closure to learned routing via closed-form OLS computation. Our key finding is that geometric moment matching—aligning query-key distributions via gauge-coupled whitening—provides no improvement over symmetric k-means. Instead, routing quality depends on directly predicting attention mass per bucket, which GWR Linear accomplishes without iterative training.

The remaining 12.2 percentage point gap to learned routing confirms that non-linear patterns exist that linear prediction cannot capture. Future work could explore non-linear training-free predictors, joint optimization across multiple heads, or integration with KV cache compression methods for further efficiency gains.

References

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

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Léon Bottou, Zhihao Jia, and Beidi Chen. Magicpig: Lsh sampling for efficient llm generation. *ArXiv*, abs/2410.16179, 2024.

R. Child, Scott Gray, Alec Radford, and I. Sutskever. Generating long sequences with sparse transformers. *ArXiv*, abs/1904.10509, 2019.

K. Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. *ArXiv*, abs/2009.14794, 2020.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *ArXiv*, abs/2307.08691, 2023.

Tri Dao, Daniel Y. Fu, Stefano Ermon, A. Rudra, and Christopher R'e. Flashattention: Fast and memory-efficient exact attention with io-awareness. *ArXiv*, abs/2205.14135, 2022.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *ArXiv*, abs/2001.04451, 2020.

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

Pierre-Emmanuel Mazar'e, Gergely Szilvasy, Maria Lomeli, Francisco Massa, Naila Murray, Herv'e J'egou, and Matthijs Douze. Inference-time sparse attention with asymmetric indexing. *ArXiv*, abs/2502.08246, 2025.

Aurko Roy, M. Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and I. Polosukhin. Attention is all you need. pp. 5998–6008, 2017.

Hong Wang and Kelly Wang. Maximal gauge symmetry in transformer architectures. *ArXiv*, 2026.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768, 2020.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ArXiv*, abs/2309.17453, 2023.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, G. Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 35 16:14138–14148, 2021.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

M. Zaheer, Guru Guruganesh, Kumar Avinava Dubey, J. Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. *ArXiv*, abs/2007.14062, 2020.

Zhenyu (Allen) Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *ArXiv*, abs/2306.14048, 2023.