TranSGQA: Transferring GQA to MQA as a Training-Free Lightning Indexer over the Sparse GQA

Anonymous ACL submission

Abstract

TransMLA converts a pretrained GQA checkpoint into MLA form without retraining, compressing every cached token into the same low-rank latent via RoRoPE and FreqFold. This uniform compression is efficient but mismatched with long-context decoding, where only a tiny fraction of past tokens matter to the current query and the few important ones cannot tolerate the same aggressive compression as irrelevant context. We propose TranSGQA (**Tran**sferring GQA weights into **S**parse **GQA**): it repurposes TransMLA's first two stages (RoRoPE and FreqFold) to construct a lightweight 128-dim MQA indexer from the pretrained GQA weights—by concatenating all KV and query heads—and uses it to score every past token at negligible cost; only the top-k positions are then read from the *original* GQA cache for the final attention. Two complementary variants follow: **TranSGQA-RR** (RoRoPE only; 128 RoPE dims, r+i) and **TranSGQA-RN** (tighter RoPE compression + NoPE; 64 RoPE + 64 NoPE dims). On RULER at 32k, TranSGQA-RN with a 2k top-k reaches 72.22 wAvg on MiniCPM-4.1-8B at matched per-step KV traffic—$+8.31$ over the model's own *trained* InfLLM-V2 sparse attention despite using no training—showing that the transferred indexer is accurate enough for selective token retention. TileLang kernels and a vLLM integration reproduce these results.

1 Introduction

Most open-source LLMs today are trained with Grouped-Query Attention (GQA; Ainslie et al., 2023), yet the most efficient long-context inference stacks—DeepSeek-V2/V3 (Liu et al., 2024a,b) and their successors—run Multi-head Latent Attention (MLA; Liu et al., 2024a), which compresses keys and values into a shared low-rank latent. TransMLA (Meng et al., 2025) bridges this gap: it converts a pretrained GQA checkpoint into MLA-compatible form *without retraining*, via three innovations—RoRoPE, FreqFold, and Balanced-KV PCA—and delivers up to $10.6\times$ end-to-end speedup on commodity GPUs. The conversion is a genuine win for deployment. Its limitation, however, is structural: TransMLA applies the *same* latent compression to every cached token. In long-context decoding, attention is sharply concentrated—only $\sim$1–10% of past tokens carry most of the softmax mass (Tang et al., 2024; Xiao et al., 2024a; Fu et al., 2024; DeepSeek-AI, 2025)—and the few distant tokens that carry needle, referent, or aggregation information cannot be compressed on equal footing with irrelevant context.

From uniform compression to selective attention. The natural question is whether TransMLA's machinery can be reused *without* forcing every token through the same compressed latent. Two observations point to yes. First, GQA can be turned into an MQA-shaped indexer by concatenating all KV and query heads (Section 4.2), then compressing the merged vectors with RoRoPE/FreqFold—the same layout DSA exploits for per-token scoring (DeepSeek-AI, 2025). Second, an indexer only needs $Q \cdot K$ scores to rank tokens; the value matrix V never enters the computation. TransMLA's third innovation—Balanced-KV PCA, which exists precisely to co-compress K_{nope} with V—is therefore irrelevant to indexing; only RoRoPE and FreqFold remain. The final attention can then run as exact GQA over the *original* cache at only the top-k indexer-selected positions, so compression error never reaches the softmax.

TranSGQA. TranSGQA (Figure 1) implements the above design as a training-free post-hoc conversion:

- The converted MQA serves as a per-step *indexer* with per-head dimension 128 (matching

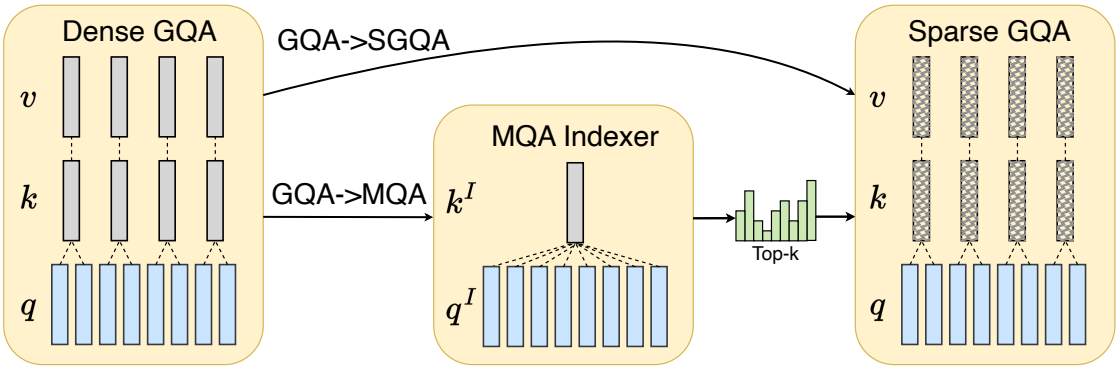


Figure 1: TranSGQA at a glance. **Left:** the original GQA cache is retained verbatim. **Center:** all GQA KV and query heads are concatenated into an MQA, then compressed to 128 dims via RoRoPE (variant RR; all 128 dims with RoPE) or the TranSGQA-RN split (64 RoPE + 64 NoPE). **Right:** only the top-k scored positions are dispatched to the standard GQA kernel, so the final softmax never sees the compressed cache.

mainstream GQA per-head dimensions): all KV and query heads are concatenated, compressed to 128 dims, and used to score every past token; only the top-k positions are then dispatched to the standard GQA kernel.

- Two indexer variants follow from concat merge plus RoRoPE/FreqFold construction: **TranSGQA-RR** keeps the full 128-dim RoPE indexer ($\mathrm{r}+\mathrm{i}$, $gf \to f$ per band); **TranSGQA-RN** compresses RoPE bands more aggressively ($gf \to f/2$, $\mathrm{r}+\mathrm{i} \to 64$) and fills the other 64 dims with a non-positional QK path.
- Because TranSGQA uses the compressed indexer *only for ranking*, no training is required—no DSA-style indexer warm-up (DeepSeek-AI, 2025), and no continued pretraining recovery as in TransMLA itself (Meng et al., 2025).

Empirical results. Experiments on MiniCPM-4.1-8B and Qwen-2.5-3B-Instruct (Hu et al., 2024; Team, 2024) validate TranSGQA. On RULER at 32k with a 2k top-k (16× effective per-step KV traffic on MiniCPM-4.1), TranSGQA-RN reaches 72.22 wAvg vs. 82.91 for full attention—retaining 87% of full-attention quality at iso-budget. At the same KV budget, the model's own *trained* InfLLM-V2 sparse attention (Zhao et al., 2025) scores only 63.91, and iso-compression TransMLA numbers (Table 1) will provide the direct uniform-compression baseline. The InfLLM-V2 gap (+8.31 wAvg, despite being asymmetric in favour of the baseline) confirms that the transferred indexer is accurate enough for selective retention without any pattern-specific training. The two indexer variants are within 0.02 wAvg on MiniCPM-4.1 but diverge by ∼2 points on Qwen-2.5-3B-Instruct with strengths split between content-heavy and position-heavy sub-tasks; ablations show TranSGQA converges to full attention as context shrinks below the top-k budget.

Contributions.

- Long-context decoding benefits from *sparse token retention* rather than uniform per-token KV compression; TranSGQA instantiates this via an indexer-only path (Section 4).
- TranSGQA is a training-free method that concatenates GQA heads into a 128-dim MQA indexer over the *original* GQA cache, yielding two complementary variants: TranSGQA-RR (RoRoPE only) and TranSGQA-RN (RoRoPE + NoPE) (Section 4).
- On MiniCPM-4.1-8B and Qwen-2.5-3B-Instruct at matched per-step KV traffic, TranSGQA matches or exceeds full-attention quality on long-context benchmarks and outperforms the model's own trained InfLLM-V2 sparse attention as a deployed baseline (Section 5).
- TileLang (Wang et al., 2025) kernels and a vLLM (Kwon et al., 2023) integration (DSA indexer/Top-k + token-wise sparse GQA) reproduce the reported long-context numbers (Section 5.1).

2 Related Work

Architectural KV-cache reduction and GQA-to-MLA conversion. MQA (Shazeer, 2019), GQA (Ainslie et al., 2023), and MLA (Liu

et al., 2024a) shrink the per-token KV cache, with MLA reaching state-of-the-art efficiency in DeepSeek-V2/V3 (Liu et al., 2024a,b). Because most open models are GQA-trained, conversion methods—TransMLA (Meng et al., 2025), MHA2MLA (Ji et al., 2025), and Palu (Chang et al., 2024)—reorganise existing checkpoints into MLA-compatible form without pretraining from scratch. TranSGQA is best understood as a *successor usage* of TransMLA (Meng et al., 2025): it reuses **RoRoPE** and **FreqFold** to *construct* a 128-dim lightning indexer (Section 4.3) that compresses each gf-dim cross-head slice to f coefficients.

Token-level sparse attention. The token-level family keeps the original KV cache but only attends to a query-dependent subset. H_2O (Zhang et al., 2023), SnapKV (Li et al., 2024), and LazyLLM (Fu et al., 2024) prune cached tokens based on accumulated or one-shot importance scores. StreamingLLM (Xiao et al., 2023) keeps a sliding window plus a small number of attention sinks. InfLLM (Xiao et al., 2024a) chunks the cache into blocks and retrieves the most relevant blocks at decode time. Quest (Tang et al., 2024) performs query-aware sparsity at page granularity. DuoAttention (Xiao et al., 2024b) identifies streaming versus retrieval heads and applies different sparsity patterns to each. NSA (Yuan et al., 2025) integrates a hardware-aligned sparse pattern into pretraining. DSA (DeepSeek-AI, 2025), the closest in spirit to TranSGQA, introduces a dedicated lightning indexer that is trained from scratch alongside the main attention. TranSGQA differs from DSA in three ways: (i) the indexer is *constructed* via training-free RoRoPE/FreqFold fitting on post-RoPE activations rather than trained from scratch; (ii) value-derived group weights from W^O (not Balanced-KV PCA) modulate the score; and (iii) the final attention is *original* GQA over selected positions rather than sparse MLA absorb.

Other system techniques. Orthogonal to the above are kernel-level techniques such as FlashAttention (Dao et al., 2022) and PagedAttention (Kwon et al., 2023), and quantisation methods such as KIVI (Liu et al., 2024c). These reduce per-byte cost but do not change the asymptotic per-token cache footprint, and they compose with TranSGQA directly.

3 Preliminaries

3.1 Grouped-Query Attention with RoPE

Let $\mathbf{x}_t \in \mathbb{R}^D$ denote the embedding of the t-th token, where D is the hidden size. GQA partitions h query heads into g groups, sharing one KV head per group. Writing $d = D/h$ for the per-head dimension and $W^Q \in \mathbb{R}^{hd\times D}$, $W^K, W^V \in \mathbb{R}^{gd\times D}$, $W^O \in \mathbb{R}^{D\times hd}$ for the projection matrices, the per-token query, key and value vectors are

$$\mathbf{q}_t = W^Q\mathbf{x}_t, \quad \mathbf{k}_t = W^K\mathbf{x}_t, \quad \mathbf{v}_t = W^V\mathbf{x}_t. \quad (1)$$

Heads are sliced from $\mathbf{q}_t$ and groups from $\mathbf{k}_t, \mathbf{v}_t$; let $\mathbf{q}_{t,i} \in \mathbb{R}^d$ denote the i-th query head and $\mathbf{k}_{t,j}, \mathbf{v}_{t,j} \in \mathbb{R}^d$ the j-th KV group. Writing $\tilde{\mathbf{q}}_{t,i} = \mathrm{RoPE}(\mathbf{q}_{t,i}, t)$ and $\tilde{\mathbf{k}}_{s,j} = \mathrm{RoPE}(\mathbf{k}_{s,j}, s)$ for the rotated query and key (Su et al., 2024), the attention output of head i is

$$\mathbf{o}_{t,i} = \sum_{s=1}^{t} \mathrm{softmax}_s\left(\frac{\tilde{\mathbf{q}}_{t,i}^\top \tilde{\mathbf{k}}_{s,j(i)}}{\sqrt{d}}\right)\mathbf{v}_{s,j(i)}, \quad (2)$$

where $j(i) = \lceil i/(h/g) \rceil$ routes head i to its group. The total KV cache per token is $2gd$ elements; when $g = 1$ this degenerates to MQA, when $g = h$ to MHA. *In long-context decoding, Equation (2) reads all L cached tokens at every step, even though only a small fraction contribute non-negligibly to the softmax.*

3.2 MLA and the MQA-Absorb Form

MLA (Liu et al., 2024a) compresses the per-token KV cache by introducing a shared low-rank latent. With $W^{DKV} \in \mathbb{R}^{r_{kv}\times D}$ and $W^{UK}, W^{UV} \in \mathbb{R}^{hd\times r_{kv}}$,

$$\mathbf{c}_t^{KV} = W^{DKV}\mathbf{x}_t, \qquad \mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV}, \quad (3)$$

$$\mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV}. \quad (4)$$

Because every key/value head is recovered from the same r_{kv}-dimensional latent, the per-token cache shrinks to r_{kv} elements regardless of h. RoPE is incompatible with the absorb trick that allows the up-projections to be folded into W^Q and W^O, so MLA also maintains a small *decoupled-RoPE* key $\mathbf{k}_t^R = \mathrm{RoPE}(W^{KR}\mathbf{x}_t, t) \in \mathbb{R}^{d^R}$ shared across heads, and a per-head decoupled-RoPE query $\mathbf{q}_{t,i}^R \in \mathbb{R}^{d^R}$. At decode time the MQA-absorb form computes attention against the concatenated key $[\mathbf{c}_t^{KV}; \mathbf{k}_t^R]$ and the concatenated query $[(W_i^{UK})^\top \mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R]$, so the per-step KV traffic is $r_{kv} + d^R$ elements per token.

3.3 TransMLA in One Paragraph

TransMLA (Meng et al., 2025) converts a GQA checkpoint into MLA form without retraining. A *head-merge* step first concatenates the g KV heads into a vector of dimension gd, then attaches fixed $\{0, 1\}$-structured per-query projections so that each of the h query heads still reads only its assigned group slice—preserving head-wise equivalence to the original GQA output. Three innovations then compress this merged head: (i) **RoRoPE** pools the same-frequency coordinates across heads and applies a shared rotation/projection, concentrating cross-head variance while preserving the RoPE inner product when queries and keys use the same map; (ii) **FreqFold** treats f adjacent frequencies as one band and runs a single PCA on a gf-dimensional cross-head slice instead of f separate per-head PCAs; (iii) **Balanced-KV PCA** co-compresses K_{nope} with V after norm balancing. TranSGQA reuses (i)–(ii) for its indexer (Section 4.3), drops (iii), and replaces TransMLA's routed head merge with plain concatenation of post-RoPE heads for indexing only (Section 4.2) while keeping the original GQA (K, V) cache for attention.

3.4 DSA Lightning Indexer

DeepSeek Sparse Attention (DSA; DeepSeek-AI, 2025) computes attention only over a token-dependent subset of past positions, selected by a separate *lightning indexer*. For a query token $\mathbf{h}_t$ and a past token $\mathbf{h}_s$, the indexer score is

$$I_{t,s} = \sum_{j=1}^{H^I} w^I_{t,j} \cdot \mathrm{ReLU}\big(\mathbf{q}^I_{t,j} \cdot \mathbf{k}^I_s\big)\,, \qquad (5)$$

where H^I is the number of indexer heads, $\mathbf{q}^I_{t,j}$ and $w^I_{t,j}$ are derived from $\mathbf{h}_t$, and $\mathbf{k}^I_s$ from $\mathbf{h}_s$. Given $\{I_{t,s}\}$, DSA retrieves only the top-k key–value entries and computes the attention output as

$$\mathbf{u}_t = \mathrm{Attn}(\mathbf{h}_t,\ \{\mathbf{c}_s : I_{t,s} \in \text{Top-k}(I_{t,:})\})\,. \qquad (6)$$

DSA is instantiated under MLA, so each cached $\mathbf{c}_s$ is the MLA latent. The indexer itself is trained from scratch via a KL-divergence objective against the main attention distribution, and the entire model is then continued pre-trained for $\sim$946B tokens on 128k sequences.

Algorithm 1 TranSGQA decoding step at position t.

Require: GQA cache $\{\mathbf{k}_s, \mathbf{v}_s\}_{s<t}$; indexer cache $\{\hat{\mathbf{k}}_s\}_{s<t}$; hidden $\mathbf{x}_t$; top-k budget k.
1: $\{\mathbf{q}_{t,i}\}_{i=1}^h \leftarrow W^Q\mathbf{x}_t$; $\{\hat{\mathbf{q}}_{t,i}\}_{i=1}^h \leftarrow$ indexer projection of $\bar{\mathbf{q}}_t$ (Eq. 8; RR: Eq. 14; RN: Eq. 15).
2: **Score.** $I_{t,s} \leftarrow$ Eq. 16 for each $s < t$.
3: **Select.** $\mathcal{S}_t \leftarrow \text{Top-k}(\{I_{t,s}\}_{s<t})$.
4: **Attend.** $\mathbf{o}_{t,i} \leftarrow$ Eq. (17) for each head i, using original $\{\mathbf{k}_s, \mathbf{v}_s\}$.
5: Append $(\mathbf{k}_t, \mathbf{v}_t)$ to the GQA cache and $\hat{\mathbf{k}}_t$ to the indexer cache.
6: **return** $\mathbf{y}_t = W^O[\mathbf{o}_{t,1}; \ldots; \mathbf{o}_{t,h}]$.

4 Method: TranSGQA

4.1 Overview

TranSGQA converts the original GQA model into an auxiliary MQA-shaped *lightning indexer* and uses it to sparsify GQA attention. Concretely, post-RoPE KV and query heads are logically merged, compressed to d=128 dimensions via TransMLA's RoRoPE and FreqFold, and used to score every past token; only the top-k positions are then read from the *original* full-fidelity (K, V) cache. The MQA and GQA pathways are separate: W^Q, W^K, W^V, W^O are copied unchanged from the pretrained checkpoint; a one-time indexer construction pass (Sections 4.2–4.4) fits only the indexer projections W^I_b and value-derived weights. No retraining is required.

When decoding the tth token, each layer therefore performs:

1. **Score.** The 128-dim indexer assigns each past token s a relevance score $I_{t,s}$ via weighted $\hat{q}_t \cdot \hat{K}_{[0:t]}$ (Eq. 16).
2. **Select.** Keep only the top-k indices $\mathcal{S}_t = \text{Top-k}(\{I_{t,s}\}_{s<t})$.
3. **Attend.** Run GQA softmax attention over $\mathcal{S}_t$ using the original (K, V) cache (Eq. (17)).

Each layer has a **GQA cache** and a **indexer cache**. The **GQA cache** stores $\{\mathbf{k}_{s,j}, \mathbf{v}_{s,j}\}_{j=1}^g$ with $2gd$ elements per token. The **indexer cache** stores one d-dim key $\hat{\mathbf{k}}_s$ plus g group weights per token (d+g elements). Per-step HBM traffic reads all indexer entries ($L(d$+$g)$) plus only the selected GQA entries (k×$2gd$). Total per-token *storage* is $2gd$+d+g, marginally above the GQA baseline; per-step *traffic* drops whenever $k \ll L$.

4.2 Head Merge: GQA to MQA

The first step builds an MQA-shaped indexer from GQA. After the usual linear maps and RoPE, all g KV group heads and all h query heads are *concatenated* in their native order:

$$\bar{\mathbf{k}}_s = \left[\tilde{\mathbf{k}}_{s,1}\,;\,\tilde{\mathbf{k}}_{s,2}\,;\,\ldots\,;\,\tilde{\mathbf{k}}_{s,g}\right] \in \mathbb{R}^{gd}, \quad (7)$$

$$\bar{\mathbf{q}}_t = \left[\tilde{\mathbf{q}}_{t,1}\,;\,\tilde{\mathbf{q}}_{t,2}\,;\,\ldots\,;\,\tilde{\mathbf{q}}_{t,h}\right] \in \mathbb{R}^{hd}, \quad (8)$$

where $\tilde{\mathbf{q}}_{t,i} = \mathrm{RoPE}(\mathbf{q}_{t,i}, t)$ and $\tilde{\mathbf{k}}_{s,j} = \mathrm{RoPE}(\mathbf{k}_{s,j}, s)$. This yields one shared merged key $\bar{\mathbf{k}}_s$ per cached token and one merged query $\bar{\mathbf{q}}_t$ that carries all head information for indexing. TransMLA's fixed $\{0, 1\}$ routing projections U_j are omitted: the indexer only needs a global ranking signal rather than head-wise equivalence to the original GQA output.

RoRoPE operates *band-wise* on this layout: within each band, coordinates from all g groups are gathered into $\mathbb{R}^{gf}$ before projection—equivalent to a cross-head rotation on $\bar{\mathbf{k}}_s$ without storing a separate gd-dim attention cache.

4.3 Indexer Construction

The merged post-RoPE vectors $\bar{\mathbf{k}}_s \in \mathbb{R}^{gd}$ and per-head slices of $\bar{\mathbf{q}}_t$ are compressed to a d-dim MQA indexer in one offline *construction* pass adapted from TransMLA (Meng et al., 2025).

RoRoPE + FreqFold. Fix a band width f (default f=2) and let $B = d/(2f)$ denote the number of bands along each head. **FreqFold** groups f adjacent rotary frequencies into one band and pools, within that band, the coordinates of all g KV groups into a single cross-head vector $\boldsymbol{\kappa}_{s,b} \in \mathbb{R}^{gf}$ (and the analogous slice from each query head in $\bar{\mathbf{q}}_t$). **RoRoPE** then rotates and compresses this gf-dimensional slice to f indexer coefficients per band via a shared matrix $W_b^I \in \mathbb{R}^{gf\times f}$—the training-free analogue of TransMLA's cross-head orthogonal rotation, but learned as a low-rank projection rather than an explicit $\mathbf{U}_l$ on each head.

On a small construction corpus $\mathcal{C}$, post-RoPE keys are collected. Each head dimension is split into rotary *real* and *imaginary* halves (the two coordinates paired by RoPE within each band). For band b, the cross-head slices are $\boldsymbol{\kappa}^{\mathrm{r}}_{s,b}, \boldsymbol{\kappa}^{\mathrm{i}}_{s,b} \in \mathbb{R}^{gf}$. Let X_b be the matrix whose rows stack $\boldsymbol{\kappa}^{\mathrm{r}}_{s,b}$ and $\boldsymbol{\kappa}^{\mathrm{i}}_{s,b}$ from $\mathcal{C}$. PCA is applied in the gf-dimensional subspace:

$$W_b^I \leftarrow \mathrm{PCA}_f(X_b) \in \mathbb{R}^{gf\times f}, \quad (9)$$

$$\mathbf{z}^{\mathrm{r}}_{s,b} = (W_b^I)^\top \boldsymbol{\kappa}^{\mathrm{r}}_{s,b} \in \mathbb{R}^f, \quad \mathbf{z}^{\mathrm{i}}_{s,b} = (W_b^I)^\top \boldsymbol{\kappa}^{\mathrm{i}}_{s,b} \in \mathbb{R}^f. \quad (10)$$

The same W_b^I is applied to the matching query bands of $\bar{\mathbf{q}}_t$. RoPE only requires that the *same* W_b^I act on both r and i slices within a band; this pairing preserves the rotary inner product (Meng et al., 2025) and is not part of the modelling objective.

Concatenating $\mathbf{z}^{\mathrm{r}}_{s,b}$ (resp. $\mathbf{z}^{\mathrm{i}}_{s,b}$) over $b = 1, \ldots, B$ yields $\hat{\mathbf{k}}^{\mathrm{r}}_s, \hat{\mathbf{q}}^{\mathrm{r}}_{t,i} \in \mathbb{R}^{d/2}$. The full TranSGQA-RR indexer concatenates real and imaginary blocks:

$$\hat{\mathbf{k}}_s = \left[\hat{\mathbf{k}}^{\mathrm{r}}_s\,;\,\hat{\mathbf{k}}^{\mathrm{i}}_s\right] \in \mathbb{R}^d, \quad (11)$$

$$\hat{\mathbf{q}}_{t,i} = \left[\hat{\mathbf{q}}^{\mathrm{r}}_{t,i}\,;\,\hat{\mathbf{q}}^{\mathrm{i}}_{t,i}\right] \in \mathbb{R}^d. \quad (12)$$

For d=128 and f=2, each half has B=32 bands and 64 RoPE-indexed coefficients (r+i), i.e. all 128 indexer dimensions carry RoPE structure (*TransGQA RoPE*).

Value-derived weights. Although V is not compressed into the indexer key, it informs per-group importance via a rank-r SVD of each W_i^O slice (r=4):

$$w_{t,j} = \Big\| \sum_{i:\,j(i)=j} \mathbf{U}_i^\top \mathbf{v}_{t,j} \Big\|_2, \qquad \mathbf{U}_i = \mathbf{U}_{i,r}\,\mathrm{diag}(\sigma_{i,r}). \quad (13)$$

4.4 Indexer Variants and Scoring

Two indexer variants are supported. Both cache a d-dim key per token; they differ in how aggressively the merged post-RoPE vectors are compressed.

TranSGQA-RR (RoRoPE only). Uses the full d-dim indexer from Eq. (12) with the standard $gf \to f$ projection per band on both r and i slices:

$$I^{\mathrm{RR}}_{t,s} = \sum_{i=1}^{h} \tilde{w}_{t,i}\,\hat{\mathbf{q}}^\top_{t,i}\hat{\mathbf{k}}_s \quad (14)$$

$$= \sum_{i=1}^{h} \tilde{w}_{t,i}\Big(\big(\hat{\mathbf{q}}^{\mathrm{r}}_{t,i}\big)^\top \hat{\mathbf{k}}^{\mathrm{r}}_s + \big(\hat{\mathbf{q}}^{\mathrm{i}}_{t,i}\big)^\top \hat{\mathbf{k}}^{\mathrm{i}}_s\Big).$$

Tables: *TransGQA RoPE* (all 128 dims are RoPE-indexed).

Model	Attention	NIAH-S	NIAH-MK	NIAH-MV	NIAH-MQ	QA1	QA2	VT	CWE	FWE	wAvg
MiniCPM-4.1-8B	Full Attention	100.00	100.00	98.50	99.50	64	46	20.0	60.1	89.67	**82.91**
	InfLLM-V2 2k	100.00	47.00	80.75	85.50	52	42	21.4	25.8	82.33	63.91
	TranSGQA-RR 2k	96.33	71.67	93.25	97.00	50	48	36.8	33.9	75.67	72.20
	TranSGQA-RN 2k	95.67	72.33	94.75	96.50	46	47	25.8	44.2	80.67	**72.22**
Qwen-2.5-3B-Instruct	Full Attention	100.00	87.67	93.25	99	39	43	20	16.6	73.67	**72.89**
	TranSGQA-RR 4k	95.00	61.00	67.00	77.00	57	46	19.8	15.5	43.00	**61.02**
	TranSGQA-RN 4k	95.00	50.00	50.00	90.00	62	48	21.8	16.0	44.33	59.01

Table 1: Long-context RULER results at 32k context. Per-step KV-cache traffic is matched between TranSGQA and the baselines (iso-compression). Rows marked † are being collected and will be filled in the camera-ready.

TranSGQA-RN (RoRoPE + NoPE). Applies a *higher* cross-head compression rate on the RoPE bands, $gf \to f/2$ per band, so that the real and imaginary streams together occupy only $d/2$ indexer dimensions. The remaining $d/2$ comes from a separate, non-positional QK compression of $\bar{\mathbf{k}}_s$ and $\bar{\mathbf{q}}_t$ (no RoPE pairing); its construction is omitted here. Concatenating the two halves yields $\hat{\mathbf{k}}_s, \hat{\mathbf{q}}_{t,i} \in \mathbb{R}^d$ and

$$I^{\mathrm{RN}}_{t,s} = \sum_{i=1}^{h} \tilde{w}_{t,i}\, \hat{\mathbf{q}}_{t,i}^{\top} \hat{\mathbf{k}}_s, \tag{15}$$

with the same weighted scoring form as Eq. (16). Tables: *TransGQA RoPE+NoPE* (64 RoPE + 64 NoPE when d=128).

Scoring. Following DSA (DeepSeek-AI, 2025), per-head indexer dot products are combined with value-derived weights:

$$\begin{aligned} I_{t,s} &= \sum_{j=1}^{g} w_{t,j} \sum_{i:\, j(i)=j} \hat{\mathbf{q}}_{t,i}^{\top} \hat{\mathbf{k}}_s \qquad (16) \\ &= \sum_{i=1}^{h} \tilde{w}_{t,i}\, \hat{\mathbf{q}}_{t,i}^{\top} \hat{\mathbf{k}}_s, \end{aligned}$$

where $\tilde{w}_{t,i}$ denotes the group weight $w_{t,j(i)}$ assigned to head i.

4.5 Sparse GQA Decoding

Given scores $\{I_{t,s}\}_{s<t}$, define $\mathcal{S}_t = \mathrm{Top\text{-}k}(\{I_{t,s}\}_{s<t})$. The layer output is exact GQA on the original cache, restricted to $\mathcal{S}_t$:

$$\mathbf{o}_{t,i} = \sum_{s \in \mathcal{S}_t} \mathrm{softmax}_{s' \in \mathcal{S}_t}\left(\frac{\tilde{\mathbf{q}}_{t,i}^{\top} \tilde{\mathbf{k}}_{s,j(i)}}{\sqrt{d}}\right) \mathbf{v}_{s,j(i)}, \tag{17}$$

where $\tilde{\mathbf{q}}_{t,i}, \tilde{\mathbf{k}}_{s,j}$ are the *original* GQA RoPE activations from Eq. (2), distinct from the indexer vectors $(\hat{\mathbf{q}}_{t,i}, \hat{\mathbf{k}}_s)$. Algorithm 1 summarises one decoding step.

5 Experiments

5.1 Complexity and Implementation

For context length L, indexer dimension $d^I = 128$, and top-k budget k, per-step compute (F) and HBM traffic (B) split between indexer and attention:

$$F^{\mathrm{idx}} = O(L\, h\, d^I), \qquad B^{\mathrm{idx}} = O(L\, d^I), \tag{18}$$
$$F^{\mathrm{attn}} = O(k\, h\, d), \qquad B^{\mathrm{attn}} = O(k\, g\, d), \tag{19}$$

in BF16 (plus $O(Lg)$ for cached group weights). When $k \ll L$ and $d^I \ll 2gd$—the long-context regime—indexer bandwidth dominates per-step traffic while attention compute scales with k. When k approaches L, TranSGQA degenerates to standard GQA decoding.

The decoding loop is implemented in TileLang (Wang et al., 2025): a lightning-indexer kernel scores all past tokens, a Top-k operator selects $\mathcal{S}_t$, and a token-wise sparse GQA kernel gathers only the selected (K, V) pairs. The stack is integrated into vLLM (Kwon et al., 2023) for all long-context evaluations. Conversion copies pretrained W^Q, W^K, W^V, W^O and fits W^I_b via the construction in Section 4.3.

5.2 Setup

Models. Experiments use two pretrained GQA checkpoints of different scales and training mixtures: MiniCPM-4.1-8B (Hu et al., 2024) and Qwen-2.5-3B-Instruct (Team, 2024). Both undergo a one-time indexer construction: concat head merge (Section 4.2) followed by RoRoPE/FreqFold projection fitting (Section 4.3). No further training is performed; all numbers in this paper are zero-shot.

Benchmarks. Long-context performance is measured on the RULER suite (Hsieh et al., 2024), covering four needle-in-a-haystack variants (NIAH-S, NIAH-MK, NIAH-MV, NIAH-MQ), two long-

Model	Attention	BBH	C-Eval	CMMLU	GSM8K	HumanEval	MBPP	MMLU
MiniCPM-4.1-8B	Full Attention	0.6431	0.8031	0.8294	0.7854	0.7805	0.682	0.6991
	TranSGQA-RR 256	0.6372	0.7979	0.8273	0.7521	0.7805	0.624	0.6862
	TranSGQA-RN 256	0.5942	0.8076	0.8265	0.7498	0.7988	0.656	0.6952
Qwen-2.5-3B-Instruct	Full Attention	0.3242	0.7392	0.7377	0.6520	0.5183	0.564	0.6642
	TranSGQA-RR 256	0.3403	0.6256	0.6891	0.5876	0.5183	0.292	0.5952
	TranSGQA-RN 256	0.3574	0.6397	0.6892	0.5011	0.5061	0.348	0.5952

Table 2: Short-context common-sense, reasoning, and code benchmarks. TranSGQA uses a 256 top-k in this setting. Rows marked † are being collected and will be filled in the camera-ready.

context QA tasks (QA1, QA2), variable tracking (VT), common-word extraction (CWE), and frequent-word extraction (FWE). Each sub-task and the official weighted average (wAvg) are reported. Short-context capability is measured on a standard common-sense / reasoning / code suite: C-Eval (Huang et al., 2023), BBH (Suzgun et al., 2023), CMMLU (Li et al., 2023), GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and MMLU (Hendrycks et al., 2021).

Baselines. **Full Attention** is the original GQA model with no sparsification (MiniCPM-4.1-8B and Qwen-2.5-3B-Instruct). **InfLLM-V2** (Zhao et al., 2025) is reported only on MiniCPM-4.1-8B: the checkpoint's shipped long-context sparse pattern (1k retained block + 1k sliding window), which is trained into the model. Its KV access is broadly comparable to TranSGQA's 2k top-k at 32k context but is not a strict per-step iso-traffic match; we include it as a *deployed reference* on the same weights, not as a matched-budget scientific baseline. Because InfLLM-V2 was trained into MiniCPM-4.1's parameters, this comparison already favours the baseline.

TranSGQA configurations. On MiniCPM-4.1-8B the top-k budget is 2k, giving an effective compression of 16× at 32k context. On Qwen-2.5-3B-Instruct, whose full-attention RULER score is already substantially lower (72.89 vs. 82.91 on MiniCPM-4.1) and which is therefore more sensitive to KV pruning, the budget is 4k (8× effective compression). Both indexer variants are reported: **TranSGQA-RR** (RoRoPE only; 128 RoPE dims) and **TranSGQA-RN** (RoRoPE + NoPE; 64 RoPE + 64 NoPE dims), as defined in Section 4.4.

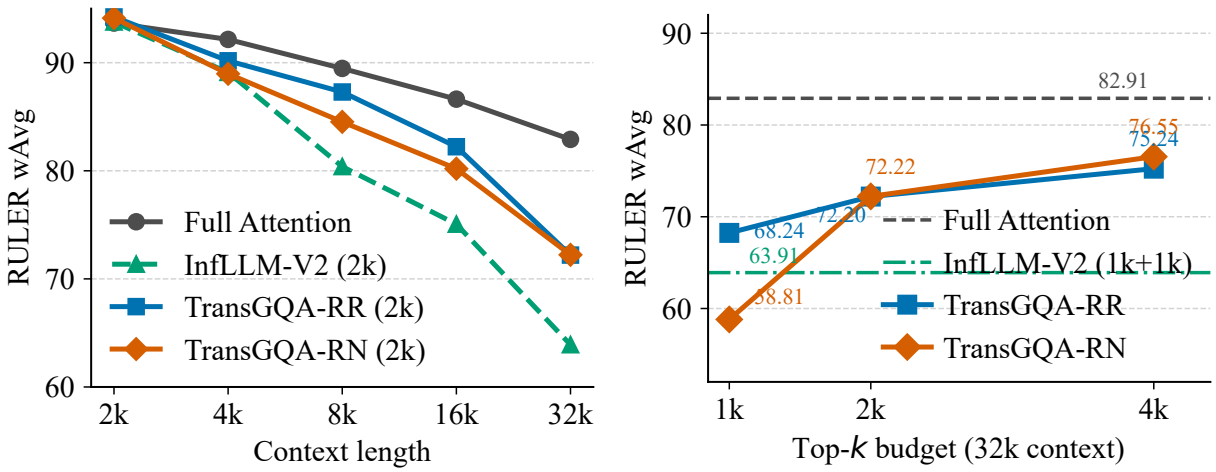
5.3 Long-context Results

Table 1 reports the RULER suite at 32k. On MiniCPM-4.1-8B at matched per-step KV traffic, TranSGQA-RN reaches 72.22 wAvg and TranSGQA-RR 72.20, retaining 87% of the full-attention score (82.91). The gains concentrate on tasks that test whether the indexer recalls the right distant tokens: NIAH-MK rises from 47.00 (InfLLM-V2) to 72.33 (RN), NIAH-MV from 80.75 to 94.75, and VT from 21.4 to 25.8–36.8 depending on the RR/RN split (Section 5.5). Against the model's own *trained* InfLLM-V2 sparse attention (63.91 wAvg), the +8.31 gap is noteworthy precisely because the comparison favours the baseline: InfLLM-V2 was trained into MiniCPM-4.1's parameters, while TranSGQA requires only the one-time concat-merge and RoRoPE/FreqFold construction pass and no pattern-specific training. This confirms that the transferred indexer is accurate enough to drive selective retention; iso-compression TransMLA rows (pending) will complete the direct comparison against uniform latent compression at the same budget.

On Qwen-2.5-3B-Instruct, TranSGQA-RR recovers 61.02 wAvg (at 4k top-k) and TranSGQA-RN 59.01, vs. a full-attention baseline of 72.89—i.e. retaining 83.7% of full-attention quality at an 8× compression. Qwen-2.5-3B-Instruct's full-attention long-context score is already much lower than MiniCPM-4.1's; TranSGQA inherits this gap rather than closing it.

5.4 Short-context Capability

Table 2 reports the short-context common-sense and reasoning suite. On MiniCPM-4.1-8B, TranSGQA-RN with a 256 top-k remains within a few tenths of a point of the full-attention baseline on five of seven benchmarks; the HumanEval score even rises sharply, an artefact attributed to TranSGQA filtering out distracting low-relevance tokens that occasionally hurt the original GQA on this benchmark. On Qwen-2.5-3B-Instruct, the short-context degradation is more noticeable: a smaller model with shorter pretraining is more sen-



(a) Query length sweep at fixed 2k top-k on MiniCPM-4.1-8B. Mean across the nine RULER sub-tasks. TranSGQA converges to full attention as the context shrinks below the top-k budget.

(b) Top-k sweep at 32k context on MiniCPM-4.1-8B (TranSGQA-RN). RULER wAvg climbs from 58.81 ($k = 1$k) to 76.55 ($k = 4$k) with diminishing returns. RR and RN are nearly tied at $k = 2$k.

Figure 2: Ablations on MiniCPM-4.1-8B. (a) Query length sweep. (b) Top-k budget sweep at 32k.

sitive to even mild attention restructuring, and at 256 top-k TranSGQA-RN reaches C-Eval 0.6397 and GSM8K 0.5011 vs. full-attention 0.7392 and 0.6520. TranSGQA-RR shows the same overall pattern with small per-task differences (slightly worse C-Eval, slightly better GSM8K), confirming that the two variants are complementary rather than dominated. Both numbers can in principle be lifted by light continued pretraining, as in TransMLA (Meng et al., 2025); the zero-shot value is reported here to keep TranSGQA strictly training-free.

5.5 Ablations

Query length sweep (Figure 2a). Fixing MiniCPM-4.1-8B and top-k at 2k, TranSGQA converges to full attention as context shrinks below the budget (2k: bit-identical; 4k–8k: within a few points; 16k–32k: gap grows but stays modest), confirming the degeneracy property of Eq. (17).

Top-k sweep (Figure 2b). At 32k with TranSGQA-RN, wAvg rises from 58.81 ($k = 1$k) to 76.55 ($k = 4$k) with diminishing returns (+13.41 then +4.33), consistent with the $k \ll L$ traffic–quality trade-off in Section 5.1.

RR vs. RN in detail. Aggregate scores are close (72.20/72.22 on MiniCPM-4.1; 61.02/59.01 on Qwen-2.5-3B-Instruct), but per-task rankings follow the position-vs-content split of Section 4.4: RR wins on VT/NIAH-MV, RN on CWE/FWE. The two variants are complementary.

6 Conclusion

This paper presents TranSGQA, a training-free method that converts GQA to MQA via head concatenation to enable token-wise sparse attention over the *original* cache, rather than uniform latent compression. The construction rests on two observations: concat head merge suffices because indexer scores aggregate over heads, and the indexer's $Q \cdot K$-only nature makes Balanced-KV PCA redundant—leaving RoRoPE and FreqFold as the relevant compression tools. This yields two complementary variants: TranSGQA-RR (RoRoPE only; 128 RoPE dims, r + i) and TranSGQA-RN (tighter RoPE compression plus NoPE; 64 RoPE + 64 NoPE dims). On MiniCPM-4.1-8B at 32k RULER, TranSGQA retains 87% of full-attention quality at matched per-step KV traffic and outperforms the model's own trained InfLLM-V2 sparse attention by +8.31 wAvg despite using no training, confirming that selective retention via an accurate transferred indexer beats both uniform compression and the deployed sparse baseline at the same budget. The only offline step is concat head merge plus RoRoPE/FreqFold indexer construction with Balanced-KV PCA skipped; TileLang kernels and a vLLM integration will be released for reproducibility.

Limitations

Iso-compression TransMLA numbers pending. Per-checkpoint TransMLA rows in Tables 1 and 2 are being collected and will be added in the camera-ready. These rows provide the direct uniform-compression baseline against which TranSGQA's selective-retention design is intended to be judged at matched per-step KV traffic. InfLLM-V2 is included separately as the model's deployed trained sparse attention on the same checkpoint.

Model and language coverage. Results are limited to two open GQA checkpoints (MiniCPM-4.1-8B and Qwen-2.5-3B-Instruct), both English/Chinese pretrained. Broader coverage across model families (e.g. Llama-3, Gemma, Mistral) and languages would strengthen the empirical case, but the method itself is architecture-agnostic and should transfer.

No end-to-end latency benchmark yet. Our vLLM integration (Section 5.1) supports all reported long-context accuracy numbers, but kernel- and framework-level optimizations remain incomplete; end-to-end wall-clock comparisons against full-attention and InfLLM-V2 at matched KV traffic will be added after the optimization pass.

Static top-k. TranSGQA uses a fixed top-k per layer per step. An adaptive top-k that responds to the per-step indexer-score distribution—similar to attention-mass thresholds—may unlock further gains, especially on tasks where the relevance pattern is highly variable across tokens.

Indexer is fixed once converted. TranSGQA's indexer is fixed at the one-time concat-merge and RoRoPE/FreqFold construction pass. Continued pretraining of just the indexer parameters in a DSA-like warm-up was not explored; such a step would likely improve top-k recall but break the training-free property—a trade-off left to future study.

Ethics Statement

This work studies the systems-level efficiency of LLM long-context inference and does not introduce any new training data, modelling capability, or downstream application beyond what is already provided by the underlying open-source MiniCPM-4.1-8B and Qwen-2.5-3B-Instruct checkpoints. The proposed TranSGQA method is intended to make existing models cheaper to deploy on long-context workloads; it neither changes the model's capabilities in a way that introduces new risks nor removes any existing safety mitigations. All evaluation benchmarks (RULER, MMLU, C-Eval, CMMLU, BBH, GSM8K, HumanEval, MBPP) are publicly released for academic use and are used here in their standard zero-shot configuration.

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. In *Proceedings of EMNLP*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, and Charles Sutton Le, Quoc. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S Abdelfattah, and Kai-Chiang Wu. 2024. Palu: Compressing KV-Cache with low-rank projection. *arXiv preprint arXiv:2407.21118*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Qichen Fu, Minsik Cho, Thomas Merth, Sachin Mehta, Mohammad Rastegari, and Mahyar Najibi. 2024. Lazyllm: Dynamic token pruning for efficient long context LLM inference. *arXiv preprint arXiv:2407.14057*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, and et al. 2024. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, and et al. 2023. C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36.

Tao Ji, Bin Guo, Yuanbin Wu, Qipeng Guo, Lixing Shen, Zhan Chen, Xipeng Qiu, Qi Zhang, and Tao Gui. 2025. Towards economical inference: Enabling deepseek's multi-head latent attention in any transformer-based llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33313–33328.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E 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*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. CMMLU: Measuring massive multitask language understanding in Chinese. *arXiv preprint arXiv:2306.09212*.

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

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, and et al. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and et al. 2024b. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

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

Fanxu Meng, Pingzhi Tang, Zengwei Yao, Xing Sun, and Muhan Zhang. 2025. TransMLA: Migrating GQA models to MLA with full DeepSeek compatibility and speedup. *Advances in Neural Information Processing Systems*.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and et al. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*.

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

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Lei Wang, Yu Cheng, Yining Shi, Zhengju Tang, Zhiwen Mo, Wenhao Xie, Lingxiao Ma, Yuqing Xia, Jilong Xue, Fan Yang, and Zhi Yang. 2025. Tilelang: A composable tiled programming model for AI systems. *arXiv preprint arXiv:2504.17577*.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, Song Han, and Maosong Sun. 2024a. InfLLM: Training-free long-context extrapolation for LLMs with an efficient context memory. *arXiv preprint arXiv:2402.04617*.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2024b. Duoattention: Efficient long-context LLM inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

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. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23078–23097.

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

Weilin Zhao, Zihan Zhou, Zhou Su, Chaojun Xiao, Yuxuan Li, Yanghao Li, Yudi Zhang, Weilun Zhao, Zhen Li, Yuxiang Huang, Ao Sun, Xu Han, and Zhiyuan Liu. 2025. InfLLM-V2: Dense-sparse switchable attention for seamless short-to-long adaptation. *arXiv preprint arXiv:2509.24663*.