GAUGEROPE: ROPE-COMPATIBLE EXECUTABLE LATENT ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent KV cache compression reduces memory footprint by mapping historical keys and values into compact representations. However, generic compressed key states require reconstruction to native dimension before applying RoPE and computing attention scores, adding sequence-scaling overhead to the decode path. We characterize when a latent KV representation can serve directly as the execution state of pretrained RoPE. For RoPE-based transformers, exact static linear execution of the rotary key path requires the latent-to-native map to preserve the pretrained RoPE action through an intertwining relation, restricting exact mixing to coordinates that share the same RoPE frequency. Guided by this characterization, we build GAUGEROPE, a training-free executable latent KV representation. It combines a RoPE-compatible carrier that preserves the rotary action with an approximate position-independent residual key–value (RKV) correction for complementary key and value information, both evaluated without per-token dense historical K/V reconstruction. Experiments on Mistral-7B show competitive quality at $2\times$ KV-state reduction and substantially stronger RULER and LongBench performance as the state budget tightens. Additional MHA results validate the construction beyond GQA. By enabling direct latent execution, GaugeRoPE achieves $1.59\times$ the decode throughput of the vanilla uncompressed baseline on 32K-context generation with batch size 4 while halving the KV cache.

1 INTRODUCTION

Autoregressive LLM (Vaswani et al., 2017) decoding maintains a growing history of keys and values that must be stored and repeatedly accessed at every generation step. At long context lengths and larger batch sizes, this KV cache becomes a substantial source of GPU memory footprint and memory traffic. Architectural designs such as GQA (Grouped-query attention) (Ainslie et al., 2023) and MLA (Multi-head latent attention) (DeepSeek-AI et al., 2024) reduce KV state by changing the attention architecture itself. Post-training methods instead operate on an already pretrained cache through KV quantization (Liu et al., 2024; Hooper et al., 2024), token eviction (Zhang et al., 2023; Xiao et al., 2024), or low-rank compression.

Existing low-rank KV compression methods primarily optimize storage: they encode historical K/V into compact latent states, while generic compressed keys must still be reconstructed into native space for RoPE-aware scoring. Palu (Chang et al., 2025) exploits redundancy in pretrained KV states, but its RoPE-aware key path reconstructs historical keys before native positional scoring. MLA (DeepSeek-AI et al., 2024) avoids full native-key reconstruction by learning a compressed content latent together with a separate RoPE branch during pretraining, while TransMLA (Meng et al., 2025) adapts pretrained models toward this latent architecture via parameter uptraining. For pretrained GQA and MHA (Multi-head attention) models, the underlying issue is structural: unless a latent key space is closed under the pretrained rotary action, RoPE cannot be applied directly within that space. Each decode step revisits the full history, so native-key reconstruction adds overhead that scales with sequence length and erodes the benefit of compressed storage. This motivates the following question: *can the compressed latent state itself be executed directly by the pretrained attention operator?* We consider a training-free setting in which the pretrained RoPE action remains fixed and the compressed historical states are directly executable without per-token key reconstruction.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Let $\mathbf{U}$ denote the static latent-to-native key map and let $\mathbf{R}_p$ denote the pretrained positional action at position p. A generic low-rank decoder may mix coordinates rotated at different RoPE frequencies, so applying RoPE after decoding is generally not representable by a fixed latent transformation. Exact direct execution therefore requires

$$\mathbf{R}_p\mathbf{U} = \mathbf{U}\mathbf{R}'_p \qquad \text{for all positions } p, \tag{1}$$

for some latent rotation $\mathbf{R}'_p$.

We characterize the static linear latent maps that remain exactly compatible with pretrained RoPE under mild non-degeneracy assumptions. The resulting structure is frequency block-diagonal: distinct RoPE frequencies cannot mix exactly, whereas same-frequency copies across heads admit arbitrary complex-linear mixing. GQA and MHA provide such same-frequency copies across KV heads or attention heads, enabling compression within each frequency block while preserving the pretrained positional action. The same compatibility condition is necessary and sufficient for exact latent inner-product scoring in our execution model, allowing historical keys to be scored directly in latent space without materializing dense native-space keys.

Figure 1 illustrates this distinction. GaugeRoPE combines a RoPE-compatible carrier for rotary scoring with a position-independent RKV correction for residual key/value content. Both operate directly on cached latent states, avoiding per-token reconstruction of native-dimensional K/V.

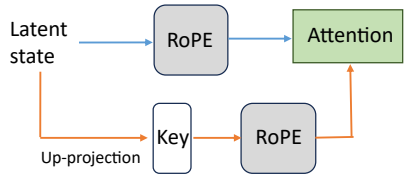

Figure 1: Why RoPE compatibility matters. Generic latent keys require native-space reconstruction before RoPE (orange), whereas GaugeRoPE supports direct latent execution (blue).

Experiments on GQA and MHA backbones evaluate the compatible construction and its practical benefits. On Mistral, GaugeRoPE preserves competitive long-context quality at $2\times$ KV-state reduction and degrades more gracefully as the state budget tightens. Its direct latent execution further improves decoding efficiency, achieving $1.59\times$ the throughput of the vanilla uncompressed baseline at 32K context and batch size 4 while halving the KV state. To isolate the effect of execution alone, we hold the cached Gauge state fixed and vary only the execution path; direct latent execution reaches $6.09\times$ the throughput of materializing native historical K/V.

Contributions.

- **RoPE-compatible latent characterization.** We characterize the static original-frequency linear maps that carry pretrained RoPE exactly, showing that distinct frequencies cannot mix while same-frequency copies admit arbitrary complex-linear mixing.
- **Executable latent attention.** Guided by this characterization, GaugeRoPE combines a RoPE-compatible carrier with a position-independent RKV correction, enabling historical attention to be evaluated directly from compressed latent states without per-token native-key reconstruction.
- **Mechanistic and systems evaluation.** Experiments disentangle the roles of compatibility, approximation capacity, and residual correction in determining quality. System measurements quantify the efficiency benefits of direct latent execution.

2 Problem Setup

2.1 RoPE as a Positional Representation

RoPE (Su et al., 2024) encodes position by rotating each two-dimensional query/key coordinate pair with a frequency-dependent phase. For one attention head with even head dimension d_h, let $F = d_h/2$ and let ω_j denote the jth RoPE frequency. Position p acts on the corresponding two-dimensional plane as

$$\mathbf{R}_j(p) = \begin{bmatrix} \cos(\omega_j p) & -\sin(\omega_j p) \\ \sin(\omega_j p) & \cos(\omega_j p) \end{bmatrix}, \qquad \mathbf{R}_p = \operatorname{diag}\big(\mathbf{R}_1(p), \ldots, \mathbf{R}_F(p)\big). \tag{2}$$

For a query at position t and a key at position s,

$$\ell_{t,s} = (\mathbf{R}_t \mathbf{q}_t)^\top (\mathbf{R}_s \mathbf{k}_s) = \mathbf{q}_t^\top \mathbf{R}_{s-t} \mathbf{k}_s, \tag{3}$$

using orthogonality and $\mathbf{R}_t^\top \mathbf{R}_s = \mathbf{R}_{s-t}$.

Identifying each two-dimensional plane with one complex coordinate makes the representation structure explicit. Write $q_j = q_{2j-1} + iq_{2j}$ and $k_j = k_{2j-1} + ik_{2j}$. Then

$$q_j \mapsto e^{i\omega_j p} q_j, \qquad k_j \mapsto e^{i\omega_j p} k_j, \tag{4}$$

and the real attention logit is

$$\ell_{t,s} = \sum_{j=1}^{F} \mathrm{Re}\left[\overline{q_{t,j}}\, e^{i\omega_j (s-t)} k_{s,j} \right]. \tag{5}$$

The complex notation is only a compact representation of ordinary real 2×2 RoPE pairs; deployment remains real-valued. Each RoPE frequency induces a distinct positional rotation. A static latent coordinate that mixes different frequencies therefore cannot itself follow one original RoPE rotation.

2.2 Why Generic Latent Keys Remain Reconstructive

Consider a generic low-rank key representation with column-vector hidden state $\mathbf{x}_s \in \mathbb{R}^d$. Using the conventional projection shape $\mathbf{W}_K \in \mathbb{R}^{d\times d_K}$, so that $\mathbf{k}_s = \mathbf{W}_K^\top \mathbf{x}_s$, write

$$\mathbf{W}_K \approx \mathbf{A}_K \mathbf{B}_K, \qquad \mathbf{A}_K \in \mathbb{R}^{d\times r_K}, \qquad \mathbf{B}_K \in \mathbb{R}^{r_K \times d_K}, \tag{6}$$

and cache the column-vector latent

$$\mathbf{h}_s^K = \mathbf{A}_K^\top \mathbf{x}_s, \qquad \hat{\mathbf{k}}_s = \mathbf{B}_K^\top \mathbf{h}_s^K. \tag{7}$$

Palu (Chang et al., 2025) is a representative post-training construction of this form.

Without a position-dependent transformation, the decoder can be absorbed into the query side:

$$\mathbf{q}_t^\top \hat{\mathbf{k}}_s = \mathbf{q}_t^\top \mathbf{B}_K^\top \mathbf{h}_s^K = \left(\mathbf{B}_K \mathbf{q}_t\right)^\top \mathbf{h}_s^K. \tag{8}$$

The value path has the same useful linearity. If $\mathbf{h}_s^V = \mathbf{A}_V^\top \mathbf{x}_s$ and $\hat{\mathbf{v}}_s = \mathbf{B}_V^\top \mathbf{h}_s^V$, then

$$\sum_s \alpha_{t,s} \hat{\mathbf{v}}_s = \mathbf{B}_V^\top \left(\sum_s \alpha_{t,s} \mathbf{h}_s^V \right), \tag{9}$$

so value decoding can occur after aggregation.

RoPE breaks the analogous absorption on the key path. The score becomes

$$(\mathbf{R}_t \mathbf{q}_t)^\top \mathbf{R}_s \mathbf{B}_K^\top \mathbf{h}_s^K, \tag{10}$$

and a generic $\mathbf{B}_K^\top$ does not admit a fixed latent action $\mathbf{R}'_s$ satisfying the corresponding intertwining relation. Implementations may fuse reconstruction, rotation, and scoring without writing the full key to off-chip memory, yet reconstruction remains part of the rotary key path. An executable latent state requires the pretrained positional action to close on the latent state itself.

3 RoPE-Compatible Latent Spaces

Same-frequency blocks. We characterize static latent spaces on which pretrained RoPE remains closed. For frequency j, stack the pre-RoPE key coordinates from all heads that share that frequency as $\boldsymbol{\kappa}_j \in \mathbb{C}^D$; thus D simply counts the same-frequency copies. In GQA (Ainslie et al., 2023), $D = H_{kv}$; in MHA, $D = H_q = H_{kv}$. All D coordinates receive the same RoPE phase $e^{i\omega_j p}$, and a rank-r_j latent block uses decoder $\mathbf{U}_j \in \mathbb{C}^{D\times r_j}$.

Theorem 1 (Exact RoPE-compatible maps)**.** *Assume the retained RoPE frequencies are pairwise distinct and no retained frequency equals the negative of another modulo 2π: $\omega_j \not\equiv \omega_k \pmod{2\pi}$ for $j \neq k$ and $\omega_j \not\equiv -\omega_k \pmod{2\pi}$ for all j, k. Let each latent block use its corresponding original RoPE frequency. Then a static linear decoder is exactly RoPE-compatible if and only if it is block diagonal in frequency: arbitrary mixing is allowed among copies of the same frequency, while mixing distinct frequencies is forbidden.*

Sketch. Write the decoder in frequency blocks. The intertwining equation forces every distinct-frequency block to vanish, while same-frequency blocks commute automatically. General real-linear maps can additionally couple conjugate frequencies; the stated assumption excludes those terms. Appendix A gives the full proof. □

Operationally, exact RoPE-compatible compression may mix across heads sharing a frequency, but cannot mix coordinates with different RoPE frequencies.

The theorem constrains executability rather than reconstruction quality. Within this compatible family, fidelity depends on retained rank and residual correction; increasing r_j adds capacity without changing the positional rule. Appendix A.2 discusses finite-window near-degeneracy.

From compatibility to direct execution. To connect the closure condition to attention, consider a semi-unitary compatible decoder $\mathbf{U}$ with $\mathbf{U}^\dagger\mathbf{U} = \mathbf{I}$. Its adjoint provides the corresponding projection into latent coordinates. A historical token at position s can cache the unrotated latent $\mathbf{z}_s = \mathbf{U}^\dagger\mathbf{k}_s$. The score against the compressed reconstruction $\mathbf{U}\mathbf{U}^\dagger\mathbf{k}_s$ is

$$\langle \mathbf{R}_t\mathbf{q}_t, \mathbf{R}_s\mathbf{U}\mathbf{U}^\dagger\mathbf{k}_s\rangle_{\mathbb{R}} = \langle \mathbf{R}'_t\mathbf{U}^\dagger\mathbf{q}_t, \mathbf{R}'_s\mathbf{z}_s\rangle_{\mathbb{R}} = \langle \mathbf{U}^\dagger\mathbf{q}_t, \mathbf{R}'_{s-t}\mathbf{z}_s\rangle_{\mathbb{R}}. \tag{11}$$

Thus the cached key remains unrotated, and historical scoring is performed directly in latent coordinates without reconstructing a native key per token.

Corollary 1 (Direct latent scoring)**.** *Under static linear encoder, decoder, and query-side maps, with the latent space taken to be the full image of the encoder, original-frequency latent rotations, and exact logit equality relative to the compressed reconstruction, exact direct latent scoring is possible if and only if the key decoder intertwines RoPE:*

$$\mathbf{R}_p\mathbf{U} = \mathbf{U}\mathbf{R}'_p \qquad \textit{for all } p. \tag{12}$$

Equation (11) gives sufficiency and Appendix A proves the converse. Thus, under this execution model, intertwining is the structural condition for a reconstruction-free fixed latent-RoPE execution path.

Proposition 1 (Rank-one phase-gauge equivalence)**.** *Every complex rank-one RoPE-compatible carrier admits an exactly equivalent phase-aligned real realization by folding per-head phases into the pretrained query and key projections. This reparameterization preserves the pretrained attention function, and the ungauged real rank-one family is a strict subset of the phase-aligned family.*

Operationally, phase alignment bridges the complex-linear characterization and real-valued inference: it implements the full complex rank-one compatible family with real operators without increasing compatible rank. Appendix A gives the construction.

4 GAUGEROPE

GaugeRoPE realizes the executable latent structure identified by the compatibility characterization. It caches historical states as two complementary components: a RoPE-compatible carrier for the rotary key path and an RKV correction for residual key and value content. The carrier preserves the pretrained RoPE action in latent space, while the RKV branch captures additional content capacity through absorbed linear maps. Together, these components enable direct execution without per-token dense historical K/V reconstruction.

Construction and execution. Offline, GaugeRoPE estimates the same-frequency carrier bases and RKV maps from calibration activations, with the resulting linear maps folded into the frozen projections where applicable. During decoding, historical tokens are stored only through their carrier coordinates and RKV codes; historical logits and value aggregation are then evaluated directly from these cached latents.

4.1 ROPE-COMPATIBLE CARRIER CONSTRUCTION

For one layer and frequency j, let $\boldsymbol{\kappa}_j(t) \in \mathbb{C}^D$ stack the D pre-RoPE key coordinates sharing that RoPE frequency at calibration token t ($D = H_{kv}$ for GQA and $D = H_q = H_{kv}$ for MHA). We

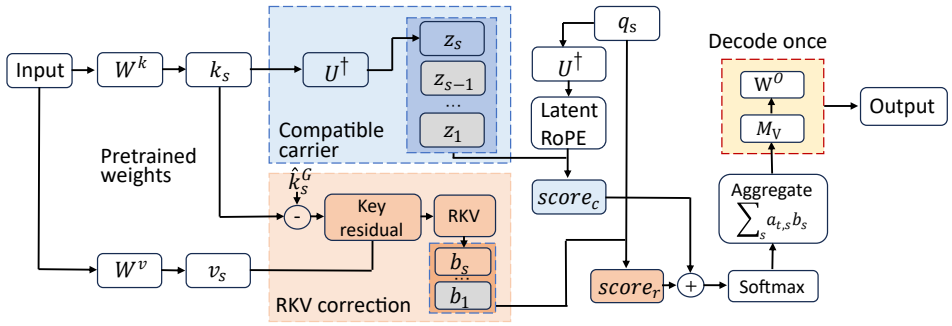

Figure 2: GaugeRoPE overview. GaugeRoPE represents each historical token with two cached latent components: a RoPE-compatible carrier $\mathbf{z}_s$ for the rotary key path and a position-independent RKV code $\mathbf{b}_s$ for complementary residual-key and value information. The two paths produce complementary attention scores that are combined before softmax, while values are aggregated in latent space and decoded once, avoiding per-token historical native-K/V reconstruction.

form the same-frequency Gram matrix

$$\mathbf{G}_j = \sum_t \boldsymbol{\kappa}_j(t)\boldsymbol{\kappa}_j(t)^\dagger \in \mathbb{C}^{D\times D}, \tag{13}$$

and use its leading eigenvectors as the carrier basis $\mathbf{U}_j$. The eigendecomposition is performed independently within the same-frequency blocks allowed by Theorem 1. Consequently, changing the retained rank changes approximation capacity but not the RoPE action carried by each latent coordinate.

For a rank-one basis, write $\mathbf{u}_j = \mathbf{D}_j\mathbf{a}_j$, where $\mathbf{D}_j$ is diagonal with the unit phases of $\mathbf{u}_j$ (assigning an arbitrary unit phase to zero entries) and $\mathbf{a}_j = |\mathbf{u}_j|$ is real. Then

$$\mathbf{u}_j^\dagger \boldsymbol{\kappa}_j = \mathbf{a}_j^\top \mathbf{D}_j^\dagger \boldsymbol{\kappa}_j. \tag{14}$$

We fold $\mathbf{D}_j^\dagger$ into the Q/K projections offline using Proposition 1, leaving a real projection at inference. This reparameterization preserves the pretrained attention function. Higher ranks use ordinary real 2×2 blocks for complex coefficients (Appendix A.5).

Here rank means the number of retained directions within a same-frequency block, not attention-operator rank. For $D = 1$, rank one already spans the full frequency block; for $D > 1$, Theorem 1 allows additional modes:

$$\mathbf{U}_j = [\mathbf{u}_{j,1}, \mathbf{u}_{j,2}, \ldots, \mathbf{u}_{j,r_j}], \qquad \mathbf{z}_{s,j} = \mathbf{U}_j^\dagger \boldsymbol{\kappa}_{s,j}. \tag{15}$$

Every added column remains in the same-frequency block and therefore keeps the same RoPE frequency. In the main GQA configuration, we use a uniform rank across frequencies: $r_j = 1$ for R1 and $r_j = 2$ for R2. For Mistral, $D = 8$, so these settings retain one or two of eight available same-frequency modes; the MHA experiments instead compare matched normalized ranks r/D. The theorem also permits frequency-dependent ranks r_j, but we use the uniform construction throughout the main experiments. For head dimension d_h, each compatible mode contributes exactly d_h real coordinates per layer and token.

4.2 Residual Key-Value Correction and Direct Attention

The compatible carrier need not span the entire key. As in decoupled-RoPE latent attention (DeepSeek-AI et al., 2024), explicit positional interaction can be concentrated in a dedicated subspace, leaving the remaining capacity for residual key and value content. Let $\hat{\mathbf{k}}^{\mathrm{G}}$ denote the key component reconstructed from the compatible carrier and define

$$\mathbf{r}^K = \mathbf{k} - \hat{\mathbf{k}}^{\mathrm{G}}. \tag{16}$$

This reconstruction is used only offline to define the calibration residual; $\hat{\mathbf{k}}^{\mathrm{G}}$ is not materialized during decoding. For the RKV latent, $\mathbf{r}^K$ and $\mathbf{v}$ are flattened over the H_{kv} heads, each in $\mathbb{R}^{H_{kv}d_h}$. After energy balancing, a joint PCA encoder $\mathbf{A}_R$ produces

$$\mathbf{b}_s = \mathbf{A}_R^\top[\mathbf{r}_s^K; \mathbf{v}_s], \qquad \hat{\mathbf{r}}_s^K = \mathbf{M}_K\mathbf{b}_s, \qquad \hat{\mathbf{v}}_s = \mathbf{M}_V\mathbf{b}_s, \tag{17}$$

where $\mathbf{M}_K$ and $\mathbf{M}_V$ are the residual-key and value blocks of the decoder. For query head g, let $h = \pi(g)$ be its shared KV head. Writing $\mathbf{e}_h \in \mathbb{C}^D$ for the corresponding coordinate in each same-frequency block, define the absorbed carrier query $\widetilde{\mathbf{q}}_{t,g,j} = \mathbf{U}_j^\dagger \mathbf{e}_h q_{t,g,j}$. The carrier score is

$$\ell^{\mathrm{G}}_{t,s,g} = \sum_j \mathrm{Re}\left[\widetilde{\mathbf{q}}^\dagger_{t,g,j} e^{i\omega_j(s-t)} \mathbf{z}_{s,j}\right], \tag{18}$$

which is the per-head GQA realization of Equation (11). For the RKV correction, let $\mathbf{S}_h \in \mathbb{R}^{d_h \times H_{kv} d_h}$ select the same KV-head block. The residual-key decoder is absorbed into the query side,

$$\ell^{\mathrm{res}}_{t,s,g} = \mathbf{q}_{t,g}^\top \mathbf{S}_h \mathbf{M}_K \mathbf{b}_s = \left(\mathbf{M}_K^\top \mathbf{S}_h^\top \mathbf{q}_{t,g}\right)^\top \mathbf{b}_s. \tag{19}$$

Historical scoring for query head g is therefore

$$\hat{\ell}_{t,s,g} = \underbrace{\ell^{\mathrm{G}}_{t,s,g}}_{\text{exact RoPE-compatible carrier}} + \underbrace{\ell^{\mathrm{res}}_{t,s,g}}_{\text{position-independent correction}}. \tag{20}$$

The carrier score is exact within the retained subspace. The RKV branch approximates the remaining position-dependent score $\mathbf{q}_{t,g}^\top \mathbf{R}_{s-t} \mathbf{S}_h \mathbf{r}_s^K$ by the position-independent $\mathbf{q}_{t,g}^\top \mathbf{S}_h \hat{\mathbf{r}}_s^K$, enabling query-side decoder absorption. Thus the overall attention is approximate even though carrier execution is exact.

Figure 2 shows the hybrid path: the carrier is scored in RoPE-compatible latent coordinates and the RKV correction is scored directly from its position-independent code before softmax. Values admit the same head-selective linear absorption. For query head g with $h = \pi(g)$,

$$\sum_s \alpha_{t,s,g} \mathbf{S}_h \hat{\mathbf{v}}_s = \mathbf{S}_h \mathbf{M}_V \left(\sum_s \alpha_{t,s,g} \mathbf{b}_s\right). \tag{21}$$

Both paths therefore avoid per-token historical native K/V reconstruction.

4.3 State Size

The theory fixes representation structure but leaves the retained capacity as a design choice, so experiments compare fixed state budgets. With per-frequency compatible rank r and a d_c-dimensional RKV code, the cached payload contains

$$d_h r + d_c \tag{22}$$

real scalars per layer and token. A full GQA KV state contains $2H_{kv}d_h$ scalars, so the nominal dimensional ratio is

$$\rho = \frac{2H_{kv}d_h}{d_h r + d_c}. \tag{23}$$

B768 and B1024 denote 768 and 1024 real scalars per historical token/layer; the main R2 configurations use $d_c = 512$ and 768, respectively.

5 Experiments

5.1 Experimental Setup

Models and budgets. We use Mistral-7B-Instruct-v0.3 as the primary GQA backbone for matched-quality, mechanism, and systems experiments. Main Mistral quality comparisons use the B768 and B1024 budgets from Section 4.3 and apply each compressed representation to every cached token. CodeLlama-7B-Instruct-hf and Gemma-7B-IT test the same construction on standard MHA.

Baselines and calibration. Full denotes the unmodified pretrained model. **Palu** (Chang et al., 2025) and **ReCalKV** (Yan et al., 2025) are external post-training low-rank references. Their compared configurations are matched to Gauge by per-token state budget. Palu and ReCalKV follow their official calibration procedures (Appendix F); Gauge uses a fixed unlabeled English corpus without evaluation prompts or labels (Appendix B). The matched unconstrained control uses a rank-256 PCA carrier and refits RKV under the same B1024 budget.

Benchmarks. RULER (Hsieh et al., 2024) is evaluated at 8K on all 13 tasks with 50 examples per task and the official metric. Mistral is additionally evaluated on all 16 English LongBench tasks (Bai et al., 2024), with native lengths capped at 32K and official task-specific metrics. For the capacity ablation, LongBench-6 denotes a six-task subset comprising PassageRetrieval-en, HotpotQA, Qasper, NarrativeQA, QMSum, and RepoBench-P.

5.2 Long-Context Quality at Matched State Budgets

Tables 1 and 2 report results at matched state budgets on the primary Mistral backbone. At $2.00\times$ reduction, GaugeRoPE achieves 85.92 RULER and 36.46 LongBench. When the budget is tightened to $2.67\times$ reduction, the gap widens substantially: Gauge B768 retains 62.33/28.84 on RULER/LongBench, compared with 47.04/23.67 for Palu and 26.49/25.31 for ReCalKV. Across these two budgets, GaugeRoPE degrades more gracefully as compression increases.

Table 1: Mistral-7B-Instruct-v0.3 RULER-8K category macros at matched state budgets. NIAH averages the eight needle-in-a-haystack tasks; CWE–FWE and QA average two tasks each.

Method	State red.	NIAH	VT	CWE–FWE	QA	Avg.
Full	$1.00\times$	95.00	98.80	93.07	68.00	90.84
Palu	$2.00\times$	91.19	92.80	75.50	55.00	83.33
ReCalKV	$2.00\times$	88.44	92.80	74.83	55.00	81.54
Gauge B1024	$2.00\times$	91.13	90.80	82.60	66.00	85.92
Palu	$2.67\times$	53.31	46.00	38.53	31.00	47.04
ReCalKV	$2.67\times$	15.75	68.80	39.80	35.00	26.49
Gauge B768	$2.67\times$	60.13	85.20	76.03	46.00	62.33

Table 2: Mistral-7B-Instruct-v0.3 LongBench results at the same matched state budgets as Table 1. Results are aggregated over 16 English tasks with a 32K input cap by the official task families.

Method	State red.	Single-doc QA	Multi-doc QA	Summ.	Few-shot	Synthetic	Code	Avg.
Full	$1.00\times$	41.00	39.42	29.30	46.37	51.25	59.81	43.15
Palu	$2.00\times$	38.69	38.06	26.21	44.65	33.00	45.05	37.43
ReCalKV	$2.00\times$	36.97	32.22	26.46	40.88	41.75	48.77	36.91
Gauge B1024	$2.00\times$	34.88	32.06	28.10	32.94	45.50	54.20	36.46
Palu	$2.67\times$	25.20	24.37	18.55	33.19	9.25	28.13	23.67
ReCalKV	$2.67\times$	27.94	20.42	20.27	31.14	16.75	36.09	25.31
Gauge B768	$2.67\times$	29.11	17.12	24.38	31.07	34.33	43.87	28.84

Appendix C.1 additionally reports a $W = 256$ rolling-window variant, which substantially improves LongBench quality relative to whole-history compression, raising B1024 from 36.46 to 38.84 and B768 from 28.84 to 35.73; the corresponding configurations retain $1.94\times$ and $2.53\times$ effective state reduction at 8K, respectively. We also evaluate a no-uptraining TransMLA conversion under the same state budgets; its category-level results are reported in Appendix F.4.

Capacity and branch complementarity. At fixed B1024 state, increasing the compatible carrier from R1 to R2 produces a sharp quality transition, raising RULER from 6.36 to 85.92 and the representative LongBench-6 macro from 0.031 to 45.63. With the R2 state fixed, neither the carrier nor the residual score branch alone preserves closed-loop behavior, indicating that sufficient compatible capacity and the RKV correction play complementary roles (Appendix D).

5.3 Extension to Standard Multi-Head Attention

The same construction extends directly to standard MHA, where $D = H_q = H_{kv}$. We evaluate CodeLlama-7B-Instruct-hf ($D = 32$) and Gemma-7B-IT ($D = 16$) at normalized compatible ranks $\alpha = r/D \in \{1/4, 1/8, 1/16\}$, using a fixed 4096-scalar latent payload for compressed history and retaining the most recent $W = 256$ tokens in native K/V form. CodeLlama achieves RULER-8K macro scores of 69.50, 67.78, and 58.80 at $\alpha = 1/4, 1/8, 1/16$, respectively, compared with 79.57 for Full. Gemma achieves 56.59, 48.12, and 29.73, respectively, compared with 74.17 for Full.

CodeLlama largely saturates by $\alpha = 1/8$, gaining only $+1.72$ from $1/8$ to $1/4$, whereas Gemma continues to benefit from additional compatible capacity, gaining $+8.48$ over the same interval. These results indicate that the same compatible construction applies under MHA, while the finite compatible capacity required for high fidelity remains backbone dependent (Appendix E).

5.4 Compatible versus Unconstrained Latent States

Table 3: Compatible versus unconstrained carrier at the matched B1024 state budget. LongBench reports the official 16-task macro.

Carrier	State red.	RULER-8K	LongBench
Gauge R2 + RKV	2.00×	85.92	36.46
Generic R256 + refit RKV	2.00×	78.20	37.80

Table 3 compares Gauge R2 with a full-space rank-256 PCA carrier at B1024 under the same state budget, with the RKV code refit for the generic carrier.

Gauge scores 7.73 points higher on RULER, while the generic carrier scores 1.34 points higher on the LongBench task macro. The quality ordering is therefore benchmark dependent, but the execution distinction is structural: Gauge retains a closed latent RoPE action, whereas the generic carrier must return to native coordinates before RoPE scoring. Paired diagnostics are reported in Appendix C.

5.5 Execution Consequences

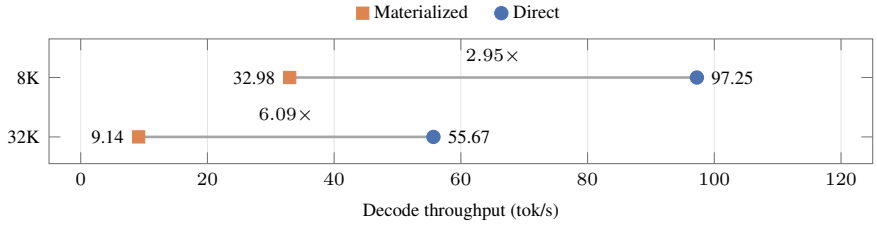

Figure 3: Same-state execution comparison (Batch=4). Direct and Materialized use the same Gauge B1024 cached state and approximation; only the execution path changes.

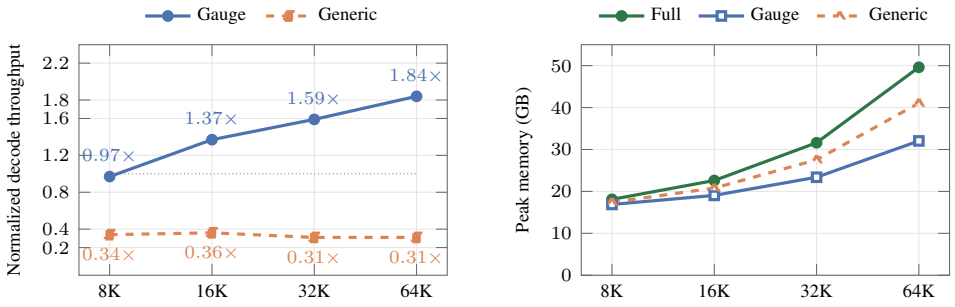

Figure 4: Decode scaling with context length (Batch=4). Left: throughput normalized by the vanilla uncompressed baseline (dotted 1× line). Right: peak allocated decode memory, including model and runtime state. Full is the vanilla uncompressed baseline and uses the native-KV path; Gauge and Generic-refit use PyTorch/GEMM reference paths.

Same state, different execution. We compare two execution paths from the same Gauge B1024 cached state. Direct performs historical attention directly from the cached latents, whereas Materialized reconstructs the corresponding native-dimensional historical K/V and then performs native-space attention. The two paths implement the same approximate operator up to numerical and accumulation differences; a parity check yields below 0.8% relative logit L2 difference and identical final argmax. At batch size 4, Direct reaches $2.95\times$ and $6.09\times$ the throughput of Materialized at 8K and 32K, respectively (Figure 3).

Decode scaling. Figure 4 shows that Gauge approaches the decode throughput of the vanilla uncompressed baseline at 8K, then reaches $1.37\times$, $1.59\times$, and $1.84\times$ that throughput at 16K, 32K, and 64K, respectively. At 64K, peak allocated memory decreases from 49.61 GB for the vanilla uncompressed baseline to 32.04 GB for Gauge, while the persistent historical KV state is halved.

Matched quality–execution trade-off. At B1024, Figure 4 compares Gauge with the unconstrained generic carrier used in Table 3. The resulting throughput gap widens with context length, from $2.87\times$ at 8K to $6.03\times$ at 64K for batch size 4 ($5.17\times$ at 32K). At 64K/B4, Gauge also reduces peak allocated memory from 41.07 GB to 32.04 GB. Raw measurements are reported in Appendix G.

6 Related Work

Latent attention and architecture conversion. MLA (DeepSeek-AI et al., 2024; DeepSeek-AI, 2024) learns a compressed content latent together with a separate RoPE branch during pretraining. TransMLA (Meng et al., 2025) and MHA2MLA (Ji et al., 2025) adapt pretrained attention toward MLA-style representations. TransMLA further introduces RoRoPE, a RoPE-aware reparameterization within this conversion. GaugeRoPE keeps the pretrained RoPE action fixed and characterizes the full static original-frequency linear family that remains closed under that action, together with its connection to direct latent scoring.

Low-rank and structured KV compression. Palu (Chang et al., 2025) is our primary training-free low-rank reference: its RoPE-aware key path reconstructs native-dimensional historical keys before positional scoring. Eigen Attention (Saxena et al., 2024) obtains per-head low-rank Q/K subspaces by SVD without RoPE-frequency constraints and likewise returns compressed keys to native dimension before rotation. ReCalKV (Yan et al., 2025), StiefAttention (Benfenati et al., 2026), STAR-KV (Bhatnagar et al., 2026), ShadowKV (Sun et al., 2025), and xKV (Chang et al., 2026) improve low-rank fidelity or systems efficiency through calibration, learned projectors, reconstruction, sparse selection, or cross-layer redundancy. GaugeRoPE targets direct executability under the pretrained RoPE operator, allowing historical attention to be evaluated from cached latent states without native-key reconstruction.

RoPE-aware and complementary cache compression. RAP (Xin et al., 2026) preserves RoPE coordinate pairs through structured pruning. From our perspective, this corresponds to coordinate selection within the compatible family, whereas our characterization additionally permits arbitrary complex-linear mixing across equal-frequency head copies. CommVQ (Li et al., 2025) instead enforces a commuting codebook interface, addressing a discrete quantization setting outside our static-linear characterization. SALS (Mu et al., 2025) targets the same RoPE-induced reconstruction bottleneck, but uses RoPE-free query–key interactions in latent space to select a sparse subset of tokens and reconstructs only the selected states, rather than maintaining a directly executable RoPE-compatible carrier. Other work explores RoPE-frequency selection (Zhou et al., 2025) and data-dependent low-rank compressibility (Chen et al., 2026), while KIVI (Liu et al., 2024), KVQuant (Hooper et al., 2024), Atom (Zhao et al., 2024), and QServe (Lin et al., 2025) primarily reduce numerical precision.

7 Conclusion

GaugeRoPE characterizes and realizes a class of executable latent states under pretrained RoPE. The RoPE-compatible carrier preserves the rotary key path within the exact compatible family, while the RKV correction provides complementary residual-key and value information through position-independent absorbed maps. Ablations separate the role of RoPE compatibility from finite-rank capacity and residual correction: compatibility enables direct execution, while capacity and correction determine the retained quality. GaugeRoPE preserves strong RULER and LongBench quality under substantial KV-state reduction, while its direct-latent implementation improves long-context decode throughput. More broadly, post-training KV compression should consider both compact storage and direct executability under the pretrained attention operator, avoiding historical native-key reconstruction.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, wording refinement, and presentation. All AI-assisted text was reviewed by the authors, who take full responsibility for the content of the manuscript.

REPRODUCIBILITY STATEMENT

Complete proofs and theoretical assumptions are provided in Appendix A. Appendix B specifies calibration data and construction details, Appendix F documents baseline reproduction protocols, and Appendix G reports system configurations, timing boundaries, and raw measurements. The supplementary material includes our source code for GaugeRoPE construction and the reported evaluations.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Luca Benfenati, Matteo Risso, Andrea Vannozzi, Ahmet Caner Yuzuguler, Lukas Cavigelli, Enrico Macii, Daniele Jahier Pagliari, and Alessio Burrello. Don't be so stief! learning KV cache low-rank approximation over the stiefel manifold, 2026. URL https://arxiv.org/abs/2601.21686.

Priyansh Bhatnagar, Ashkan Moradifirouzabadi, Se-Hyun Yang, SeungJae Lee, Jungwook Choi, and Mingu Kang. STAR-KV: Low-rank KV cache compression via soft thresholding for adaptive rank control. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://arxiv.org/abs/2606.08382. Spotlight.

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. Palu: KV-cache compression with low-rank projection. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7da6e0e00702c60607a6ae05c802ef85-Abstract-Conference.html.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Hung-Yueh Chiang, Yash Akhauri, Xilai Dai, Huiqiang Jiang, Yucheng Li, Luis Ceze, Kai-Chiang Wu, and Mohamed S. Abdelfattah. xKV: Cross-layer KV-cache compression via aligned singular vector extraction. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=XDb69lykA8.

Jian Chen, Zhuoran Wang, Jiayu Qin, Ming Li, Meng Wang, Changyou Chen, Yin Chen, Qizhen Weng, and Yirui Liu. KV-CoRE: Benchmarking data-dependent low-rank compressibility of KV-caches in LLMs, 2026. URL https://arxiv.org/abs/2602.05929.

DeepSeek-AI. DeepSeek-V3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao, Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

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

Tao Ji, Bin Guo, Yuanbin Wu, Qipeng Guo, Lixing Shen, Zhan Chen, Xipeng Qiu, Qi Zhang, and Tao Gui. 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)*, pp. 33313–33328, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1597. URL https://aclanthology.org/2025.acl-long.1597/.

Junyan Li, Yang Zhang, Muhammad Yusuf Hassan, Talha Chafekar, Tianle Cai, Zhile Ren, Pengsheng Guo, Foroozan Karimzadeh, Colorado Reed, Chong Wang, and Chuang Gan. CommVQ: Commutative vector quantization for KV cache compression. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36831–36845. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25du.html.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. QServe: W4A8KV4 quantization and system co-design for efficient LLM serving. *Proceedings of Machine Learning and Systems*, 7, 2025.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bz.html.

Fanxu Meng, Pingzhi Tang, Zengwei Yao, Xing Sun, and Muhan Zhang. TransMLA: Migrating GQA models to MLA with full DeepSeek compatibility and speedup. In *Advances in Neural Information Processing Systems*, volume 38, pp. 81977–82019, 2025. doi: 10.52202/085713-2741. URL https://papers.nips.cc/paper_files/paper/2025/hash/75d13a472f570755af2a4ae4ac3d6724-Abstract-Conference.html.

Junlin Mu, Hantao Huang, Jihang Zhang, Minghui Yu, Tao Wang, and Yidong Li. SALS: Sparse attention in latent space for KV cache compression. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/00a0ebcad584c59dbc439c2af8793638-Abstract-Conference.html.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for KV cache compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15332–15344, 2024. doi: 10.18653/v1/2024.findings-emnlp.899. URL https://aclanthology.org/2024.findings-emnlp.899/.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Hanshi Sun, Li-Wen Chang, Wenlei Bao, Size Zheng, Ningxin Zheng, Xin Liu, Harry Dong, Yuejie Chi, and Beidi Chen. ShadowKV: KV cache in shadows for high-throughput long-context LLM inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57355–57373. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25b.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

Jihao Xin, Tian Lyu, David Keyes, Hatem Ltaief, and Marco Canini. RAP: KV-cache compression via RoPE-aligned pruning, 2026. URL https://arxiv.org/abs/2602.02599.

Xianglong Yan, Zhiteng Li, Tianao Zhang, Linghe Kong, Yulun Zhang, and Xiaokang Yang. ReCalKV: Low-rank KV cache compression via head reordering and offline calibration, 2025. URL https://arxiv.org/abs/2505.24357.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1506. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6ceefa7b15572587b78ecfcebb2827f8-Abstract.html.

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

Yuhao Zhou, Sirui Song, Boyang Liu, Zhiheng Xi, Senjie Jin, Xiaoran Fan, Zhihao Zhang, Wei Li, and Xuanjing Huang. EliteKV: Scalable KV cache compression via RoPE frequency selection and joint low-rank projection, 2025. URL https://arxiv.org/abs/2503.01586.

Appendix guide. Appendix A expands the theory in Section 3; Appendix B gives calibration and construction details for Section 4; Appendices C–D provide the quality controls and mechanism diagnostics underlying Sections 5.2 and 5.4; Appendix E expands the MHA study; Appendix F documents baseline protocols and structural references; and Appendix G gives the system measurement protocols and raw execution controls.

A PROOFS

Organization. We keep the notation explicit because three linear spaces appear: the pre-RoPE key space, the compressed latent space, and the query-side space used after absorption. The proofs establish three facts used by the main text: static original-frequency RoPE-compatible maps are exactly same-frequency intertwiners; rank-one complex carriers have an exact phase-gauged real realization; and exact direct-latent execution under the stated static-map assumptions is equivalent to the intertwining condition.

RoPE as a complex representation. For each RoPE frequency j, one two-dimensional real rotation pair is identified with one complex coordinate. A real pair (x_{2j-1}, x_{2j}) is written as $x_j = x_{2j-1} + ix_{2j}$, and position p acts by multiplication with $e^{i\omega_j p}$. The real Euclidean inner product is

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{R}} = \sum_j \mathrm{Re}(\overline{x_j} y_j) = \mathrm{Re}(\mathbf{x}^\dagger \mathbf{y}).$$

Thus the complex notation below is only a compact way to write real linear algebra on RoPE rotation pairs.

Non-degeneracy assumption. Let $\chi_j(p) = e^{i\omega_j p}$. We assume that the characters are pairwise distinct and that no retained character equals the conjugate of any retained character:

$$\omega_j \not\equiv \omega_k \pmod{2\pi} \quad (j \neq k), \qquad \omega_j \not\equiv -\omega_k \pmod{2\pi} \quad (\text{all } j, k).$$

The second condition also covers $j = k$ and therefore excludes self-conjugate frequencies ($2\omega_j \equiv 0 \pmod{2\pi}$). These are algebraic identities of position characters, not assumptions about a large numerical gap between nearby frequencies. If frequencies are exactly tied, the tied components merge into a larger multiplicity space.

A.1 PROOF OF THEOREM 1

Proof. Step 1: define the source and latent representations. The stacked pre-RoPE key space is

$$\mathcal{V} = \bigoplus_{j=1}^{F} \mathcal{V}_j, \qquad \mathcal{V}_j \cong \mathbb{C}^D, \qquad D = L_g H_{kv}.$$

Here L_g denotes the number of repeated representation copies grouped in the abstract construction. The reported implementation uses $L_g = 1$; cross-layer grouping is not part of the execution path evaluated in this paper. The dimension D is the multiplicity of frequency j: each grouped layer/head slot contributes one complex coordinate at that frequency. RoPE acts on $\mathcal{V}_j$ by the scalar $e^{i\omega_j p}\mathbf{I}_D$. The latent space is

$$\mathcal{V}' = \bigoplus_{j=1}^{F} \mathcal{V}'_j, \qquad \mathcal{V}'_j \cong \mathbb{C}^{r_j},$$

and it carries the original frequency j on $\mathcal{V}'_j$, namely $e^{i\omega_j p}\mathbf{I}_{r_j}$. A linear decoder basis $\mathbf{U} : \mathcal{V}' \to \mathcal{V}$ is RoPE-compatible when

$$\mathbf{R}_p \mathbf{U} = \mathbf{U} \mathbf{R}'_p \qquad \text{for every position } p.$$

Step 2: decompose the decoder into frequency blocks. Write $\mathbf{U} = (\mathbf{U}_{jk})_{j,k}$, where $\mathbf{U}_{jk} : \mathcal{V}'_k \to \mathcal{V}_j$. Taking the (j, k) block of the intertwining equation gives

$$e^{i\omega_j p}\mathbf{U}_{jk} = \mathbf{U}_{jk} e^{i\omega_k p} = e^{i\omega_k p}\mathbf{U}_{jk}.$$

Therefore

$$\left(e^{i\omega_j p} - e^{i\omega_k p}\right)\mathbf{U}_{jk} = 0 \qquad \text{for every } p.$$

If $j \neq k$, non-degeneracy gives a position p for which the scalar coefficient is nonzero, hence $\mathbf{U}_{jk} = 0$. Thus an exact-compatible map cannot mix different RoPE frequencies.

Step 3: rule out conjugate-frequency, antilinear couplings. A general real linear map between complex coordinates can contain both a complex-linear part and an antilinear part:

$$\mathbf{T}\mathbf{z} = \mathbf{A}\mathbf{z} + \mathbf{B}\bar{\mathbf{z}}.$$

Intertwining the source frequency k with the target frequency j requires

$$e^{i\omega_j p}(\mathbf{A}\mathbf{z} + \mathbf{B}\bar{\mathbf{z}}) = \mathbf{A}(e^{i\omega_k p}\mathbf{z}) + \mathbf{B}(e^{-i\omega_k p}\bar{\mathbf{z}})$$

for every position p. Matching the coefficients of $\mathbf{z}$ and $\bar{\mathbf{z}}$ gives

$$(e^{i\omega_j p} - e^{i\omega_k p})\mathbf{A} = 0, \qquad (e^{i\omega_j p} - e^{-i\omega_k p})\mathbf{B} = 0.$$

The first equation is the block condition above. The second forces $\mathbf{B} = 0$ unless $\omega_j = -\omega_k$ modulo 2π. Under the stated non-degeneracy assumption, antilinear components vanish. Hence every compatible real map is represented by a complex-linear block matrix.

Step 4: characterize the diagonal blocks. For $j = k$, the block equation becomes

$$e^{i\omega_j p}\mathbf{U}_{jj} = \mathbf{U}_{jj}e^{i\omega_j p},$$

which is automatically true because both sides multiply by the same scalar. Therefore there is no additional constraint inside a same-frequency multiplicity space:

$$\mathbf{U}_{jj} \in \mathbb{C}^{D\times r_j} \quad \text{is arbitrary.}$$

Combining the off-diagonal exclusion with the diagonal freedom gives

$$\mathrm{Hom}_{\mathrm{RoPE}}(\mathcal{V}', \mathcal{V}) \cong \bigoplus_{j=1}^{F} \mathbb{C}^{D\times r_j}. \tag{24}$$

Step 5: add the PCA/semi-unitary constraint. The theorem characterizes all static linear intertwiners. If the decoder is additionally constrained to be a PCA-style semi-unitary basis, then each full-column-rank block satisfies

$$\mathbf{U}_{jj}^{\dagger}\mathbf{U}_{jj} = \mathbf{I}_{r_j}.$$

Thus the feasible block lies on the complex Stiefel manifold $\mathrm{St}_{\mathbb{C}}(r_j, D)$. This is an optimization constraint, not an additional RoPE constraint. □

Degenerate schedules. If two frequencies are exactly tied, $\omega_j = \omega_k$, their multiplicity spaces should be combined before applying the theorem. If $\omega_j = -\omega_k$, conjugation can intertwine the two components; if $2\omega_j \equiv 0 \pmod{2\pi}$, an antilinear self-coupling is also possible. These cases enlarge the exact commutant rather than invalidate the intertwiner formulation.

A.2 Finite-window near-degeneracy and numerical conditioning

Theorem 1 concerns exact equivariance for all integer positions. This is different from approximate equivariance on a bounded window $0 \le p \le P$. Consider a complex-linear block $\mathbf{B}_{jk} : \mathcal{V}'_k \to \mathcal{V}_j$ between distinct frequencies. Its intertwining defect at position p is

$$\|e^{i\omega_j p}\mathbf{B}_{jk} - \mathbf{B}_{jk}e^{i\omega_k p}\| = 2\left|\sin\frac{(\omega_j - \omega_k)p}{2}\right| \|\mathbf{B}_{jk}\|. \tag{25}$$

Hence $P|\omega_j - \omega_k| \ll 1$ can make a forbidden cross-frequency block nearly equivariant over that finite window even though it is not an exact intertwiner. For an antilinear block $\mathbf{C}_{jk}\bar{\mathbf{z}}$, the corresponding defect is

$$\|e^{i\omega_j p}\mathbf{C}_{jk} - \mathbf{C}_{jk}e^{-i\omega_k p}\| = 2\left|\sin\frac{(\omega_j + \omega_k)p}{2}\right| \|\mathbf{C}_{jk}\|. \tag{26}$$

Thus conjugate or low-frequency characters may also support approximate antilinear couplings on a finite context. GaugeRoPE uses the exact commutant, which yields a positional rule independent of the evaluation horizon.

Near-degenerate characters also do not create an ill-conditioned frequency separation step in GaugeRoPE. The model's RoPE indices determine the frequency groups exactly, after which each carrier is estimated from its own same-frequency Gram matrix $\mathbf{G}_j$. No linear system involving $(\omega_j - \omega_k)^{-1}$ or $(\omega_j + \omega_k)^{-1}$ is solved. Numerical sensitivity of the learned carrier is therefore governed by the eigenspectrum of $\mathbf{G}_j$, not by the spacing between different RoPE frequencies. If leading eigenvalues of $\mathbf{G}_j$ are repeated or nearly repeated, individual eigenvectors need not be identifiable, but any basis of the corresponding retained eigenspace remains inside the same exact-compatible frequency block.

A.3 Proof of Proposition 1

Proof. Step 1: polar factorization of a complex rank-one carrier. Fix a frequency j and a unit carrier $\mathbf{u}_j \in \mathbb{C}^{H_{kv}}$. For each KV head h, write

$$u_{h,j} = e^{i\phi_{h,j}}|u_{h,j}|,$$

using any phase when $u_{h,j} = 0$. Define

$$\mathbf{D}_j = \mathrm{diag}(e^{i\phi_{1,j}}, \dots, e^{i\phi_{H_{kv},j}}), \qquad \mathbf{a}_j = (|u_{1,j}|, \dots, |u_{H_{kv},j}|)^\top.$$

Then $\mathbf{a}_j \in \mathbb{R}^{H_{kv}}_{\geq 0}$, $\|\mathbf{a}_j\|_2 = 1$, and $\mathbf{u}_j = \mathbf{D}_j\mathbf{a}_j$.

Step 2: write the RoPE logit with the conjugate explicitly. Let $q_{t,g,j}$ be the query coordinate of query head g at time t, and let $\kappa_{s,h,j}$ be the pre-RoPE key coordinate of KV head h at time s. If query head g attends to KV head $h = \pi(g)$, the frequency-j contribution to the real attention logit is

$$\ell_{t,s,g,j} = \mathrm{Re}\left[\overline{q_{t,g,j}}\, e^{i\omega_j(s-t)}\kappa_{s,h,j}\right]. \tag{27}$$

Step 3: show gauge invariance of every logit term. Apply the same phase change to the query and key coordinate:

$$\widetilde{q}_{t,g,j} = e^{-i\phi_{h,j}}q_{t,g,j}, \qquad \widetilde{\kappa}_{s,h,j} = e^{-i\phi_{h,j}}\kappa_{s,h,j}, \qquad h = \pi(g).$$

Substituting into Equation (27),

$$\mathrm{Re}\left[\overline{\widetilde{q}_{t,g,j}}\, e^{i\omega_j(s-t)}\widetilde{\kappa}_{s,h,j}\right] = \mathrm{Re}\left[\overline{q_{t,g,j}}\, e^{i\omega_j(s-t)}\kappa_{s,h,j}\right].$$

The equality holds term by term, hence also after summing over frequencies and heads. In a GQA model, all query heads mapped to the same KV head receive that KV head's phase at the same frequency.

Step 4: realize the gauge as an offline weight update. Multiplication by $e^{-i\phi_{h,j}}$ is the real 2×2 rotation matrix of angle $-\phi_{h,j}$ on the corresponding RoPE plane. The gauge can therefore be folded into the learned Q and K projection weights before inference. It introduces no runtime complex arithmetic and does not change the pretrained model's attention function.

Step 5: prove rank-one equivalence. Compressing the original key vector with the complex carrier gives

$$\mathbf{u}_j^\dagger\boldsymbol{\kappa}_j = (\mathbf{D}_j\mathbf{a}_j)^\dagger\boldsymbol{\kappa}_j = \mathbf{a}_j^\top\mathbf{D}_j^\dagger\boldsymbol{\kappa}_j = \mathbf{a}_j^\top\widetilde{\boldsymbol{\kappa}}_j.$$

Thus the complex rank-one projection $\mathbf{u}_j$ in the original gauge is identical to the real rank-one projection $\mathbf{a}_j$ after phase alignment. Conversely, a real carrier is recovered by choosing $\mathbf{D}_j = \mathbf{I}$. Hence the phase-aligned real and complex rank-one families are exactly gauge-equivalent:

$$\textit{phase-aligned real rank-1} \simeq_{\mathrm{gauge}} \textit{complex rank-1}.$$

Step 6: strictness relative to the ungauged real family. The ungauged real family requires the carrier to be real in the original coordinate system. A unit vector such as $\mathbf{u} = (1, i)^\top/\sqrt{2}$ has non-constant coordinate phases and cannot be represented by any real vector in that original gauge. Therefore the naive real rank-one family is a strict subset of the phase-aligned family. □

A.4 Proof of Corollary 1

Proof. Assumptions. The corollary concerns static linear compression and exact execution relative to the compressed reconstruction. The maps are

$$\mathbf{E} : \mathcal{V} \to \mathcal{Z}, \qquad \mathbf{U} : \mathcal{Z} \to \mathcal{V}, \qquad \mathbf{Q} : \mathcal{V} \to \mathcal{Z}.$$

Here $\mathbf{E}$ is the encoder, $\mathbf{U}$ is the decoder, and $\mathbf{Q}$ is the query-side map used by absorbed attention. We assume:

1. $\mathbf{E}$ has full row rank, so $\mathcal{Z} = \operatorname{im} \mathbf{E}$ is the complete latent image.
2. The decoder is restricted to this image; there are no unused latent directions outside $\operatorname{im} \mathbf{E}$.
3. The latent space uses the original-frequency rotations $\mathbf{R}'_p$.
4. Query-side equality is required for all $\mathbf{q}, \mathbf{k}, t, s$, not merely on a finite calibration set.

The reconstructed key is $\hat{\mathbf{k}}_s = \mathbf{U}\mathbf{E}\mathbf{k}_s$.

Sufficiency. Assume first that $\mathbf{U}$ is an intertwiner:

$$\mathbf{R}_p \mathbf{U} = \mathbf{U} \mathbf{R}'_p \qquad \text{for every } p.$$

Write $\mathbf{z}_s = \mathbf{E}\mathbf{k}_s$ and define $\mathbf{c}_s = \mathbf{R}'_s \mathbf{z}_s$ for the proof. Since the intertwining relation holds for the full rotation group, its adjoint form gives $\mathbf{U}^\dagger \mathbf{R}_t = \mathbf{R}'_t \mathbf{U}^\dagger$. For any query $\mathbf{q}$ at position t,

$$\begin{aligned}
\langle \mathbf{R}_t \mathbf{q}, \mathbf{R}_s \hat{\mathbf{k}}_s \rangle_{\mathbb{R}} &= \langle \mathbf{R}_t \mathbf{q}, \mathbf{R}_s \mathbf{U} \mathbf{z}_s \rangle_{\mathbb{R}} && (28)\\
&= \langle \mathbf{R}_t \mathbf{q}, \mathbf{U} \mathbf{R}'_s \mathbf{z}_s \rangle_{\mathbb{R}} && (29)\\
&= \langle \mathbf{U}^\dagger \mathbf{R}_t \mathbf{q}, \mathbf{R}'_s \mathbf{z}_s \rangle_{\mathbb{R}} && (30)\\
&= \langle \mathbf{R}'_t \mathbf{U}^\dagger \mathbf{q}, \mathbf{R}'_s \mathbf{z}_s \rangle_{\mathbb{R}} && (31)\\
&= \langle \mathbf{R}'_t \mathbf{U}^\dagger \mathbf{q}, \mathbf{c}_s \rangle_{\mathbb{R}}. && (32)
\end{aligned}$$

Here $\mathbf{c}_s$ is only a proof variable. Equivalently, the implementation stores the unrotated latent $\mathbf{z}_s$ and applies the relative latent rotation during scoring:

$$\langle \mathbf{R}'_t \mathbf{U}^\dagger \mathbf{q}, \mathbf{R}'_s \mathbf{z}_s \rangle_{\mathbb{R}} = \langle \mathbf{U}^\dagger \mathbf{q}, \mathbf{R}'_{s-t} \mathbf{z}_s \rangle_{\mathbb{R}}.$$

Thus $\mathbf{Q} = \mathbf{U}^\dagger$ provides exact absorbed scoring for any encoder $\mathbf{E}$ with latent image $\mathcal{Z}$; the semi-unitary case used by Gauge has $\mathbf{E} = \mathbf{U}^\dagger$. Neither form materializes the dense historical key $\mathbf{R}_s \hat{\mathbf{k}}_s$.

The value branch is compatible with the same execution pattern. Once the attention weights are computed from latent scores, values are aggregated linearly in latent space and decoded once after the weighted sum.

Necessity. Now suppose static maps $(\mathbf{E}, \mathbf{U}, \mathbf{Q})$ execute exactly against compressed reconstructions:

$$\langle \mathbf{R}_t \mathbf{q}, \mathbf{R}_s \mathbf{U}\mathbf{E}\mathbf{k} \rangle_{\mathbb{R}} = \langle \mathbf{R}'_t \mathbf{Q}\mathbf{q}, \mathbf{R}'_s \mathbf{E}\mathbf{k} \rangle_{\mathbb{R}} \tag{33}$$

for every $\mathbf{q}, \mathbf{k}, t, s$. Set $t = s = 0$. Then

$$\langle \mathbf{q}, \mathbf{U}\mathbf{E}\mathbf{k} \rangle_{\mathbb{R}} = \langle \mathbf{Q}\mathbf{q}, \mathbf{E}\mathbf{k} \rangle_{\mathbb{R}}.$$

Because $\mathbf{E}$ is onto $\mathcal{Z}$, every $\mathbf{z} \in \mathcal{Z}$ can be written as $\mathbf{z} = \mathbf{E}\mathbf{k}$ for some $\mathbf{k}$. Therefore

$$\langle \mathbf{q}, \mathbf{U}\mathbf{z} \rangle_{\mathbb{R}} = \langle \mathbf{Q}\mathbf{q}, \mathbf{z} \rangle_{\mathbb{R}} \qquad \text{for all } \mathbf{q}, \mathbf{z},$$

which forces $\mathbf{Q} = \mathbf{U}^\dagger$.

Substitute $\mathbf{Q} = \mathbf{U}^\dagger$ into Equation (33). Using orthogonality of $\mathbf{R}_p$ and $\mathbf{R}'_p$, write the equality for relative offset $r = s - t$:

$$\langle \mathbf{q}, \mathbf{R}_r \mathbf{U}\mathbf{E}\mathbf{k} \rangle_{\mathbb{R}} = \langle \mathbf{q}, \mathbf{U}\mathbf{R}'_r \mathbf{E}\mathbf{k} \rangle_{\mathbb{R}} \qquad \text{for all } \mathbf{q}, \mathbf{k}, r.$$

Since this holds for all $\mathbf{q}$,

$$(\mathbf{R}_r \mathbf{U} - \mathbf{U}\mathbf{R}'_r)\mathbf{E}\mathbf{k} = 0 \qquad \text{for all } \mathbf{k}, r.$$

Using again that $\mathbf{E}$ is onto $\mathcal{Z}$, we obtain

$$\mathbf{R}_r\mathbf{U} = \mathbf{U}\mathbf{R}'_r \qquad \text{on all of } \mathcal{Z}, \text{ for every } r.$$

This is exactly the intertwining condition.

Scope. The converse characterizes static linear encoder, decoder, and query-side maps with original-frequency latent rotations and exact equality relative to the compressed reconstruction. Appendix A.2 separately quantifies the finite-window defect that arises when distinct RoPE characters are only approximately separated over a bounded context. □

A.5 GQA QUERY REALIZATION AND HIGHER-RANK REALIFICATION

For query head g, let $h = \pi(g)$ be its shared KV head and let $\mathbf{e}_h \in \mathbb{C}^D$ select that coordinate in a same-frequency block. With $\mathbf{z}_{s,j} = \mathbf{U}_j^\dagger \boldsymbol{\kappa}_{s,j}$, define

$$\widetilde{\mathbf{q}}_{t,g,j} = \mathbf{U}_j^\dagger \mathbf{e}_h \, q_{t,g,j}. \tag{34}$$

Then the carrier reconstruction obeys

$$\mathrm{Re}\Big[\overline{q_{t,g,j}} e^{i\omega_j(s-t)} \mathbf{e}_h^\top \mathbf{U}_j \mathbf{z}_{s,j}\Big] = \mathrm{Re}\Big[\widetilde{\mathbf{q}}_{t,g,j}^\dagger e^{i\omega_j(s-t)} \mathbf{z}_{s,j}\Big], \tag{35}$$

which is the GQA form of absorbed carrier scoring.

Proposition 1 gives the stronger phase-gauged real-scalar realization only at rank one. For $r_j > 1$, including the reported R2 carrier, we use the standard realification of a complex-linear block. Each complex coefficient $a + ib$ acts through

$$a + ib \quad \longleftrightarrow \quad \begin{bmatrix} a & -b \\ b & a \end{bmatrix}. \tag{36}$$

The cached complex coefficients are stored as their real and imaginary parts, matching the ordinary real-scalar cache representation used by the implementation.

B CALIBRATION AND CONSTRUCTION DETAILS

Theorem 1 fixes the admissible same-frequency family; calibration selects a finite-rank basis within that family and constructs the residual/value approximation without model training. All Gauge variants use the same fixed unlabeled English calibration corpus, stored as `calib.txt` and prepared from the WikiText-103 raw training split. The file contains 300,109 whitespace-delimited words, but calibration budgets are defined in model tokens rather than words and therefore depend on the backbone tokenizer. Carrier spectra are estimated from approximately 200K model tokens (default token cap 200,000), while the joint residual/value RKV basis uses 100K model tokens unless otherwise stated. The corpus is read sequentially in 2048-token blocks and reused across backbones and Gauge variants. No RULER or LongBench prompts, labels, answers, or evidence positions are used for Gauge calibration.

Carrier moments. Carrier spectra are estimated from the same-frequency Gram matrices in Equation (13), independently for each layer ($L_g = 1$). No cross-layer grouping is used in the reported models.

Multiplicity scope. The main text defines the per-layer multiplicities used for GQA and MHA. For MQA ($H_{kv} = 1$), rank one already spans the full within-layer compatible block, so head multiplicity alone provides no further rotary-key reduction. Cross-layer grouping could create additional same-frequency multiplicity, but we do not evaluate that extension.

Residual/value moments and energy balancing. After the carrier is fixed, we subtract its reconstructed key component,

$$\mathbf{r}^K(t) = \mathbf{k}(t) - \hat{\mathbf{k}}^{\mathrm{G}}(t). \tag{37}$$

Let $m = H_{kv} d_h$ and $\mathbf{x}(t) = [\mathbf{r}^K(t); \mathbf{v}(t)] \in \mathbb{R}^{2m}$. From the raw covariance $\mathbf{C}_0 = \mathbb{E}[\mathbf{x}\mathbf{x}^\top]$, define the per-coordinate branch energies

$$e_k = \frac{\mathrm{tr}((\mathbf{C}_0)_{:m,:m})}{m}, \qquad e_v = \frac{\mathrm{tr}((\mathbf{C}_0)_{m:,m:})}{m}, \tag{38}$$

and

$$\mathbf{S} = \mathrm{diag}\Big(e_k^{-1/2}\mathbf{I}_m,\ e_v^{-1/2}\mathbf{I}_m\Big). \tag{39}$$

We eigendecompose $\widetilde{\mathbf{C}} = \mathbf{S}\mathbf{C}_0\mathbf{S}$ and retain its leading $\mathbf{W}_c \in \mathbb{R}^{2m \times d_c}$. The cached code and reconstruction are

$$\mathbf{b} = \mathbf{W}_c^\top \mathbf{S}\mathbf{x}, \qquad \hat{\mathbf{x}} = \mathbf{S}^{-1}\mathbf{W}_c\mathbf{b}. \tag{40}$$

Thus $\mathbf{A}_R = \mathbf{S}\mathbf{W}_c$ in Equation (17), while the decoder is $\mathbf{D}_R = \mathbf{S}^{-1}\mathbf{W}_c = [\mathbf{M}_K^\top\ \mathbf{M}_V^\top]^\top$. For GQA scoring and value readout, the selector $\mathbf{S}_h$ in Equations (19) and (21) extracts the KV-head block shared by the query head.

C Quality Controls and Robustness

C.1 Rolling-Window Variant

The main experiments compress every cached token. As an optional deployment variant, we also retain the most recent $W = 256$ tokens in native K/V form while storing earlier tokens with the same Gauge latent representation. If S and B denote the native and latent per-token payloads in the same storage units, respectively, then at sequence length T the effective per-token state is

$$B_{\text{eff}}(T, W) = \frac{(T - W)B + WS}{T}, \tag{41}$$

with effective state reduction

$$\rho_{\text{eff}}(T, W) = \frac{S}{B_{\text{eff}}(T, W)} = \frac{TS}{(T - W)B + WS}. \tag{42}$$

Thus the native recent tail reduces the realized compression at finite context, while its fractional cost vanishes as T grows. For fixed 8K RULER inputs, $W = 256$ changes the nominal B1024/B768 reductions from $2.00\times/2.67\times$ to $1.94\times/2.53\times$, respectively. Tables 4 and 5 report the corresponding quality results using a common $W = 256$ effective-budget accounting. Gauge actually retains a native recent window; Palu and ReCalKV keep their whole-history dumped-model execution paths, so only their reported state-reduction labels follow this accounting.

Table 4: Rolling-window RULER results. Mistral-7B-Instruct-v0.3 RULER-8K category macros with Gauge retaining a $W = 256$ native recent window. State reduction follows the common $W = 256$ effective-budget accounting. NIAH averages the eight needle-in-a-haystack tasks; CWE–FWE and QA average two tasks each.

Method	State red.	NIAH	VT	CWE–FWE	QA	Avg.
Full	1.00×	95.00	98.80	93.06	68.00	90.84
Palu	1.94×	91.19	92.80	75.50	55.00	83.33
ReCalKV	1.94×	88.44	92.80	74.84	55.00	81.54
Gauge B1024	1.94×	92.25	91.20	85.87	63.00	86.69
Palu	2.53×	53.31	46.00	38.53	31.00	47.04
ReCalKV	2.53×	15.75	68.80	39.80	35.00	26.49
Gauge B768	2.53×	78.75	95.20	81.13	45.00	75.19

Relative to the whole-history Gauge results in Section 5.2, the $W = 256$ variant changes B1024 from 85.92 to 86.69 on RULER and from 36.46 to 38.84 on LongBench. At the tighter B768 budget, it changes RULER from 62.33 to 75.19 and LongBench from 28.84 to 35.73, while reducing the realized state reduction from the nominal $2.67\times$ to approximately $2.53\times$ for the reported $W = 256$ configuration.

Table 5: Rolling-window LongBench results. Mistral-7B-Instruct-v0.3 results with Gauge retaining a $W = 256$ native recent window. Results are aggregated over 16 English tasks with a 32K input cap by the official task families. State reduction follows the same accounting at an 8K reference length; the realized Gauge reduction varies with input length.

Method	State red.	Single-doc QA	Multi-doc QA	Summ.	Few-shot	Synthetic	Code	Avg.
Full	1.00×	41.00	39.43	29.30	46.37	51.25	59.82	43.15
Palu	1.94×	38.69	38.05	26.21	44.64	33.00	45.06	37.43
ReCalKV	1.94×	36.97	32.22	26.47	40.88	41.75	48.77	36.91
Gauge B1024	1.94×	36.58	36.90	28.52	36.70	44.50	58.18	38.84
Palu	2.53×	25.20	24.36	18.55	33.19	9.25	28.13	23.67
ReCalKV	2.53×	27.95	20.42	20.28	31.14	16.75	36.08	25.31
Gauge B768	2.53×	34.41	34.40	26.44	38.99	30.84	53.64	35.73

C.2 Does Compatibility Help at Matched State?

Section 5.4 reports the aggregate comparison. Table 6 gives the category-level breakdown for the same matched B1024 runs. The RULER paired bootstrap difference $\mathrm{Generic} - \mathrm{Gauge}$ is -7.73 with 95% CI $[-10.55, -4.88]$.

Table 6: Matched-carrier control. RULER-8K category breakdown; differences are Generic minus Gauge.

Category	Gauge	Generic	Δ
NIAH	91.13	84.31	−6.81
CWE–FWE	82.60	71.23	−11.37
QA	66.00	53.00	−13.00
VT	90.80	93.60	+2.80

For LongBench, the official task-macro difference is +1.34 points for the generic carrier, while the pooled paired example difference is -0.53 with 95% CI $[-1.47, 0.46]$. The task macro weights tasks equally, whereas pooling weights examples equally. This interval describes the pooled statistic and does not establish significance for the task-macro gap.

D Compatible Capacity and Hybrid Mechanism

D.1 What Does an Additional Compatible Mode Recover?

At fixed Mistral B1024 state, Figure 5 compares Naive-real R1, phase-gauged Gauge R1, and two-mode Gauge R2; all three use the whole-history setting at $2.00\times$ state reduction, and R1/R2 use RKV widths 896/768 while keeping total state fixed. Naive-real R1 is a realization control for the phase-gauge construction, not an unconstrained low-rank carrier. LongBench-6 contains 1200 examples total.

The closed-loop results show a sharp capacity transition. The phase-gauged R1 carrier gives only a small RULER gain over Naive R1, while both R1 variants remain near zero on LongBench-6. In contrast, R2 produces a large gain on both benchmarks. The second compatible mode adds about eight points of trace-weighted same-frequency capture. Under fixed Full teacher states, R2 also improves broad-distribution fidelity for both carrier-only and hybrid scoring, with consistent reductions in KL and score-only attention-output error (Appendix D.3). R1 already preserves the dominant route, whereas R2 recovers more of the surrounding attention distribution. This improvement occurs even though the fixed budget reduces RKV width from 896 to 768. *Takeaway:* compatibility determines whether the rotary action closes on the latent state; compatible rank sets the finite-dimensional carrier capacity available within that executable family.

D.2 How Do the Carrier and Residual KV Correction Complement Each Other?

Table 7 isolates the two score branches of Mistral Gauge R2/B1024 on RULER while keeping the cached state, calibration, RKV value path, and value aggregation fixed. Only the score mask

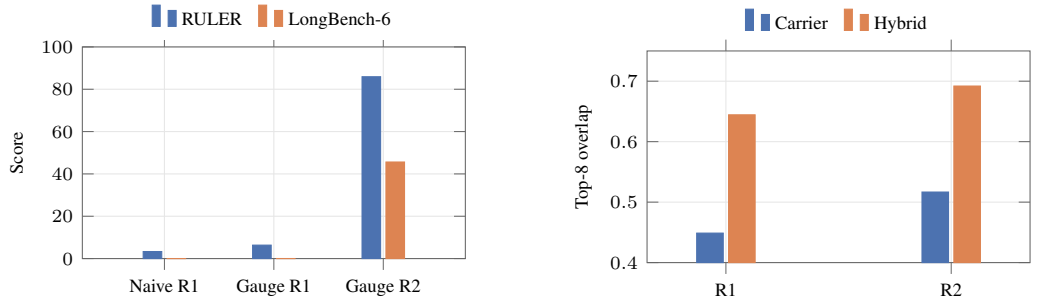

Figure 5: Compatible rank from spectrum to behavior. Left: fixed-B1024 closed-loop quality on RULER-8K and LongBench-6. LongBench-6 is the six-task representative subset used only for this carrier ablation and is not directly comparable to the full LongBench-16 macro. Right: with Full teacher states fixed, R2 improves Top-8 overlap for carrier-only and hybrid scoring. Trace-weighted same-frequency capture rises from 75.03% at C@1 to 83.06% at C@2.

changes: carrier-only uses ℓ^{G}, residual-only uses ℓ^{res}, and the full path uses their sum in Equation (20). Its teacher-forced logit statistics use a dedicated diagnostic sample.

Table 7: Compatible carrier and RKV correction are complementary. Score-branch ablation and teacher-forced historical-logit diagnostics for Mistral Gauge R2/B1024. Logit metrics compare each masked score vector with Full on the same teacher-forced states.

Score branch	RULER	MSE	KL	Top-1	Top-8 overlap	Top-2 order
Residual only	0.00	2.18×10^{-3}	1.21×10^{-4}	0.5071	0.1772	0.6342
Carrier only	4.83	5.5×10^{-5}	2.2×10^{-5}	0.9679	0.5084	0.9836
Carrier + residual	85.92	3.1×10^{-5}	1.3×10^{-5}	0.9731	0.6938	0.9866

Takeaway: the two branches are strongly complementary. Carrier-only scoring preserves the dominant teacher-forced routing order but loses broader candidate-set fidelity, while residual-only scoring lacks the position-sensitive routing structure needed for a stable rollout. Their combination improves both the score distribution and end-task quality. The diagnostic therefore matches the hybrid construction: the compatible carrier provides routing geometry, while the RKV correction supplies complementary score and value content.

D.3 Does the Extra Compatible Mode Improve the Operator Before Rollout?

Table 8 compares R1 and R2 on 32 RULER-8K prompts while fixing Full-model hidden and query states, so the measurement isolates the historical attention operator rather than rollout drift. For this diagnostic only, the old-history region excludes the most recent 256 tokens, and output drift uses Full values to isolate score/routing error from value-codec error.

Table 8: R2 improves teacher-forced operator fidelity. Old-history diagnostics for Mistral B1024; output error uses Full values.

Variant	Score form	KL ↓	Top-1 ↑	Output rel-L2 ↓
Gauge R1	Carrier only	2.6×10^{-5}	0.9605	0.0015
Gauge R1	Hybrid	1.6×10^{-5}	0.9659	0.0011
Gauge R2	Carrier only	2.0×10^{-5}	0.9645	0.0013
Gauge R2	Hybrid	1.1×10^{-5}	0.9713	0.0008

Takeaway: the operator improvement is layer-consistent: carrier-only KL and output error improve in 31/32 layers; for the hybrid score, KL improves in all 32 layers and output error in 31/32. Together with the Top-8 gains in Figure 5 and the trace-weighted capture increase $75.03\% \rightarrow 83.06\%$, this links the added compatible mode to a more faithful attention distribution before closed-loop error accumulation.

E MHA GENERALITY AT FIXED STATE BUDGET

All MHA runs retain a $W = 256$ native recent window. For the MHA sweep in Section 5.3, the carrier widths are 256/512/1024 real scalars for $\alpha \in \{1/16, 1/8, 1/4\}$, and the corresponding RKV widths are 3840/3584/3072, keeping the latent payload fixed at 4096 real scalars per compressed-history token/layer. Accounting for the native recent window, effective state reduction at 8K is $1.94\times$ for CodeLlama and $1.934\times$ for Gemma.

Section 5.3 reports the macro scores, while Table 9 gives the category-level breakdown. The paired capacity increments are: CodeLlama $+8.98$ ($1/16 \to 1/8$; 95% CI $[+5.66, +12.29]$) and $+1.72$ ($1/8 \to 1/4$; $[-0.24, +3.79]$); Gemma $+18.39$ ($[+15.17, +21.70]$) and $+8.48$ ($[+5.86, +11.20]$), respectively.

Table 9: MHA compatible capacity. Category-level RULER macros for the fixed-state-budget MHA sweep. Task Avg. is the official 13-task macro; NIAH contains eight tasks, while CWE–FWE and QA contain two each.

Backbone	Category	Full	$\alpha = 1/16$	$\alpha = 1/8$	$\alpha = 1/4$
CodeLlama	NIAH	88.94	76.19	82.81	81.75
CodeLlama	VT	94.80	16.40	86.80	87.60
CodeLlama	CWE–FWE	62.07	30.23	25.90	34.93
CodeLlama	QA	52.00	39.00	40.00	46.00
CodeLlama	Task Avg.	79.57	58.80	67.78	69.50
Gemma	NIAH	89.31	31.06	53.25	66.00
Gemma	VT	57.60	30.80	67.60	82.80
Gemma	CWE–FWE	40.07	29.57	26.97	19.47
Gemma	QA	56.00	24.00	39.00	43.00
Gemma	Task Avg.	74.17	29.73	48.12	56.59

Takeaway: the largest gains occur in NIAH and VT, whereas QA remains below Full and CWE–FWE is non-monotonic, especially on Gemma.

F BASELINE PROTOCOLS AND STRUCTURAL REFERENCES

This section serves two purposes. Sections F.1–F.2 document the reproduction protocols for Palu and ReCalKV, the matched low-rank baselines used in the main quality tables. Sections F.3–F.4 provide supplementary structural references for RAP and TransMLA, whose full pipelines involve additional training and are therefore outside our primary training-free comparison.

F.1 PALU: REPRODUCTION PROTOCOL

For Palu (Chang et al., 2025), we follow the authors' official WikiText-2 calibration procedure for whitening and low-rank decomposition. The reported Palu configurations are matched to Gauge B768 and B1024 by per-token state budget. Prompts, generation settings, task scorers, and decoding settings are shared with the corresponding main-table evaluations.

F.2 RECALKV: REPRODUCTION PROTOCOL

We reproduce ReCalKV (Yan et al., 2025) on Mistral-7B-Instruct-v0.3 using the authors' official `compress.py` pipeline and the full `decompose_method=ours` path, which uses HSR with offline value calibration and fusion. Rank search uses `fisher_uniform` with a head-group size of 4. Both calibration and updating use WikiText-2, with 256 samples per stage, a calibration sequence length of 1024, and a model sequence length of 2048. The dumped artifact contains 64 head permutation records, which we restore before evaluation with the released `configure_permutation` routine.

ReCalKV produces a projection-compressed model, and we match its two configurations to Gauge B768 and B1024 by per-token state budget. Evaluation uses the shared RULER and LongBench harness; prompts, generation settings, task scorers, and decoding settings are identical to the corresponding main-table evaluations.

F.3 RAP: RoPE-Pair Structural Reference

RAP (Xin et al., 2026) is a recent RoPE-aware structured-pruning method. We evaluate its pruning stage as a structural reference in Table 10. Q/K dimensions are retained in RoPE pairs, V is pruned channel-wise, and Fisher saliency is estimated from 32 WikiText2 batches of 1024 tokens. This is a prune-only reference: the KD/LoRA recovery stage of the full RAP pipeline is not applied. We report RAP's native retain ratio $\rho = 0.7$ and a B1024-matched $\rho = 0.5$ point. Full is re-measured in the same RAP harness.

Table 10: RAP prune-only structural reference. RULER results with Full re-measured in the same harness.

Model	Full	RAP ρ=0.7	RAP ρ=0.5
Mistral-7B-Instruct-v0.3	90.86	57.06	2.31

At the matched $\rho = 0.5$ point, quality drops sharply on Mistral; we therefore use RAP only as a prune-only structural reference rather than a matched recovery baseline.

F.4 TransMLA: No-Uptraining Structural Reference

We evaluate a post-hoc TransMLA conversion at the same state budgets used by the main quality comparisons, without uptraining or recovery updates. Prompts, generation settings, and task scorers follow the corresponding RULER and LongBench evaluations in the main text. We additionally report a short-context teacher-forced NLL evaluation for the corresponding rolling-window configurations. Table 13 reports this diagnostic, while Tables 11 and 12 report the long-context category/family aggregates.

F.4.1 Long-context category-level comparison

Table 11: No-uptraining TransMLA on RULER. Mistral-7B-Instruct-v0.3 aggregated over the four RULER categories.

Method	State red.	NIAH	VT	CWE–FWE	QA	Avg.
TransMLA	2.00×	1.50	0.00	23.07	3.00	4.93
TransMLA	2.67×	0.25	0.00	17.47	1.00	2.99

Table 12: No-uptraining TransMLA on LongBench. Mistral-7B-Instruct-v0.3 aggregated by the six official task families.

Method	State red.	Single-doc QA	Multi-doc QA	Summ.	Few-shot	Synthetic	Code	Avg.
TransMLA	2.00×	10.33	4.67	7.31	10.01	1.50	36.88	10.86
TransMLA	2.67×	7.58	2.93	5.36	7.36	1.38	33.94	8.77

F.4.2 Short-Context NLL Evaluation

Table 13: TransMLA short-context NLL evaluation. ΔNLL is relative to Full on the same 64 prompts per length.

Max tokens	Full NLL	1.94× NLL	1.94× Δ	2.53× NLL	2.53× Δ
2K	1.8445	1.9023	+0.0579	1.9203	+0.0758
4K	1.7436	1.8794	+0.1358	1.9162	+0.1725
8K	1.6823	2.3633	+0.6810	2.4258	+0.7435

G System Measurements and Execution Controls

All system measurements use one NVIDIA H100 GPU with PyTorch 2.4.0+cu121 and Triton 3.0.0. Gauge executes its B1024 executable latent state through the PyTorch/GEMM direct-execution implementation: historical scores are evaluated directly from the cached carrier coordinates and RKV

codes, latent values are aggregated before the final linear decode, and native-dimensional historical K/V are never materialized. Model loading, offline calibration, and artifact construction are excluded.

Table 14: Raw matched execution scaling. B1024 compatible-versus-generic measurements. Throughput is decode tokens/s; memory columns report peak allocated GPU memory during decode. Peak memory for the vanilla uncompressed baseline is reported for the batch-4 context sweep underlying Figure 4; "–" denotes an unreported measurement.

Length × batch	Full tok/s	Gauge tok/s	Generic tok/s	Gauge / Generic	Full peak GB	Gauge peak GB	Generic peak GB
8K × B1	55.9557	25.3570	23.9221	1.06×	–	15.2257	14.8419
8K × B4	101.7031	99.1129	34.5084	2.87×	18.1000	16.8901	17.4148
16K × B4	52.5100	71.8200	19.0400	3.77×	22.6000	19.0500	20.7900
32K × B1	25.4889	24.4587	9.3955	2.60×	–	16.7999	17.3655
32K × B4	32.2204	51.1723	9.9027	5.17×	31.6000	23.3823	27.5522
64K × B1	14.5779	22.3590	5.0592	4.42×	–	18.9015	20.7437
64K × B4	16.5608	30.5159	5.0625	6.03×	49.6100	32.0385	41.0691

How do the execution interfaces scale?

Table 14 reports the raw measurements underlying the main-text scaling figure, including matched B1024 Full/Gauge/Generic throughput and peak decode memory.

What changes when only execution form changes?

Figure 3 reports the batch-4 same-state execution comparison. The direct-execution path uses the Gauge direct-latent runner, whereas the materialized path reconstructs the historical native state before native-space scoring. The same-state execution comparison uses a separate full-model harness with BF16 state and runs 128 autoregressive decode steps. Timing includes model forward passes, cache updates, and next-token argmax. The scaling figure uses the PyTorch/GEMM direct-execution implementation.