RRC-KV: RANGE-DRIVEN ROTATION AND CENTERING FOR TWO-BIT KV CACHES

Anonymous authors
Paper under double-blind review

ABSTRACT

Scalar INT2 KV caches have only four reconstruction levels per group, making accuracy sensitive to quantization coordinates. We present RRC-KV, which learns per-head key and per-layer value rotations through range-based objectives. A query-based auxiliary regularizer limits coordinate sensitivity concentration, while static key centering shifts the encoding origin with its contribution restored in attention logits. The method requires neither downstream labels nor quantizer backpropagation and supports mixed-precision attention with fused Triton kernels. Across four models and five reasoning and coding benchmarks, RRC-KV achieves a macro-average score of 79.54 versus 80.04 for BF16. Under the same KV storage layout, it outperforms the spectral-rotation baseline OSCAR in every tested RULER setting across two models and 8K–64K contexts (87.19 versus 81.61 on average), while a long-context gap to BF16 remains. Across three Qwen models, scores on two coding tasks vary by at most 3.05 points between two calibration configurations, compared with up to 34.69 points for OSCAR on Qwen3-32B. At 64K tokens, the KV representation provides $6.91\times$ compression relative to BF16, including metadata and retained BF16 windows.

1 INTRODUCTION

The key–value (KV) cache grows with sequence length and is repeatedly read during generation. Scalar INT2 retains simple affine decoding and grouped storage, but its four reconstruction levels make resolution sensitive to large elements that widen the range. Rotations redistribute channel magnitudes while preserving unquantized computation.

Equivalent transformations underpin many low-bit methods, including channel scaling in SmoothQuant and fixed or learned rotations in QuaRot and SpinQuant (Xiao et al., 2022; Ashkboos et al., 2024; Liu et al., 2024a). KV-cache methods exploit the different statistics and roles of keys and values (Liu et al., 2024b; Hooper et al., 2024; Su et al., 2025); OSCAR constructs spectral rotations from an attention-aware covariance objective (Zhou et al., 2026). We ask: *for a fixed two-bit layout, how should we choose coordinates that support both reconstruction and attention?*

RRC-KV minimizes squared key ranges and a smooth value-range objective, with a query regularizer limiting key-side sensitivity concentration. Static key centering makes absolute-value clipping operate on deviations from the calibration mean; logits restore the center contribution. Values remain uncentered: paired diagnostics show that centering improves reconstruction but worsens attention readout.

Key residuals change attention weights; value residuals enter the weighted sum. Thus, lower reconstruction error need not yield lower output error. Our analysis links range, encoding origin, and attention error to guide representation design.

Fused Triton kernels integrate the representation into mixed-precision attention. Evaluations cover task accuracy, long contexts, component effects, storage, and throughput. Calibration stability is another empirical advantage: across Alice/RFC and GPQA configurations, RRC-KV's scores change by at most 3.05 points on two coding tasks across three Qwen models; OSCAR's changes exceed 34 points on Q32 (Section 4.4).

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

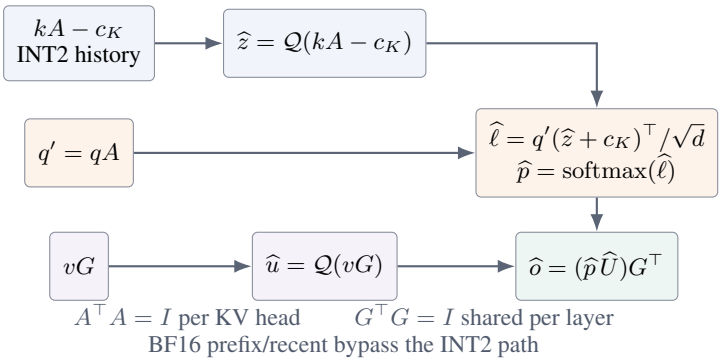


Figure 1: RRC-KV data flow. Per-head K and per-layer V rotations are fitted before key centers. Logits restore center contributions; BF16 prefix/recent tokens retain an unquantized path.

2 Attention Error in Quantization Coordinates

We use row vectors and omit the layer index. For KV head h, let $\mathcal{H}(h)$ denote its associated query heads. For $a \in \mathcal{H}(h)$ and vectors $q_{t,a}, k_s, v_s \in \mathbb{R}^{1\times d}$,

$$\ell_{t,a,s} = q_{t,a}k_s^\top/\sqrt{d}, \qquad p_{t,a} = \mathrm{softmax}(\ell_{t,a}), \qquad o_{t,a} = p_{t,a}V_t. \tag{1}$$

We partition the visible cache into low-bit history $\mathcal{L}_t$ and a BF16 region $\mathcal{B}_t$.

Computation-preserving rotations. For each KV head, choose $A \in \mathbb{R}^{d\times d}$ with $A^\top A = I$. Setting $k' = kA$ and $q' = qA$ preserves $q'(k')^\top = qk^\top$. Both transforms follow RoPE because a general orthogonal rotation does not commute with position-dependent RoPE (Su et al., 2021). Values use an orthogonal G, so $(pVG)G^\top = pV$. All these transforms preserve vector norms.

Error propagation. Let $E_{K,t}$ and $E_{V,t}$ denote encoding errors in the rotated coordinates, with zero rows for $\mathcal{B}_t$. Writing $\widehat{p} = \mathrm{softmax}(\ell + \delta\ell)$ gives the exact identities

$$\delta\ell_{t,a} = \frac{q_{t,a}AE_{K,t}^\top}{\sqrt{d}}, \qquad \widehat{o} - o = (\widehat{p} - p)V_t + \widehat{p}E_{V,t}G^\top. \tag{2}$$

The terms change attention routing and content, respectively. Unlike reconstruction error, logit error weights residuals by query second moments, so their rankings can differ. We use uncentered moments to include query means.

Coupling between representation and error. During calibration, we aggregate $C_Q = \mathbb{E}_q[q^\top q]$ over the associated queries and define $M(A) = A^\top C_Q A$. Under the product of the query and key-error marginal distributions,

$$J_K(A) = \frac{1}{d}\,\mathrm{tr}(M(A)C_E(A))\,, \qquad C_E(A) = \mathbb{E}_k[(e^K)^\top e^K]. \tag{3}$$

Here C_E depends on the rotation. If it were fixed at $\sigma^2 I$, then $J_K = \sigma^2\,\mathrm{tr}(C_Q)/d$ would be rotation-invariant. The relevant effect is therefore a change in quantization residuals, not total query energy. Appendix A.1 relates actual causal pairs to this proxy.

3 Range-Driven Rotations and Key Centering

We fit per-KV-head A and layer-shared G, then estimate c_K. At inference (Figure 1), store $\mathcal{Q}(kA - c_K)$ and $\mathcal{Q}(vG)$, restore the center in logits, and apply $G^\top$ after value aggregation; BF16 windows retain an unquantized path.

3.1 KEY CALIBRATION: RANGE-DRIVEN ROTATION LEARNING

For a rotated row $x = kA$, define $\mathrm{rng}(x) = \max_j x_j - \min_j x_j$. When the scale floor is inactive, four-level affine quantization has a grid spacing equal to one-third of the clipped row range. Let x_c be the clipped row and $\mathcal{Q}_0$ a four-level quantizer with exact metadata and nearest rounding. Then

$$\|\mathcal{Q}_0(x_c) - x_c\|_2^2 \le \frac{d}{36}\,\mathrm{rng}(x_c)^2 \le \frac{d}{36}\,\mathrm{rng}(x)^2. \tag{4}$$

Each rounding error is at most half a grid step. With clipping and metadata errors detailed in Appendix C, this motivates

$$P_{\mathrm{raw}}(A) = \mathbb{E}_k[\mathrm{rng}(kA)^2] \tag{5}$$

as a proxy for grid-error scale. Minimizing only the post-clipping range can shrink the grid while increasing clipping error. The pre-clipping range also bounds clipping deviations; squaring emphasizes rows with overly coarse grids.

Query-based auxiliary regularization. The range objective controls quantization grid scale; the query regularizer constrains the concentration of readout sensitivity across coordinates. With $m = \mathrm{diag}\, M(A)$ and $\bar{m} = d^{-1}\,\mathrm{tr}\, C_Q$, define the squared coefficient of variation

$$c_Q^2(A) = \frac{\sum_j (m_j - \bar{m})^2}{(d-1)\bar{m}^2}. \tag{6}$$

The sensitivity term is zero when query energy is zero; the implementation floors the denominator for numerical stability.

For a fixed total error budget, the worst-case diagonal allocation amplifies error by $r_m = \max_j m_j/\bar{m}$ relative to uniform allocation. Appendix B bounds this factor in terms of c_Q, motivating c_Q^2 as a constraint on sensitivity concentration. Adding it as an auxiliary regularizer gives

$$\boxed{\mathcal{L}_K(A) = P_{\mathrm{raw}}(A) + c_Q^2(A).} \tag{7}$$

Range is primary; query regularization balances squared reading weights while preserving $\mathrm{tr}\, M(A)$. We fit uncentered keys without a straight-through estimator (STE) for quantizer gradients, then estimate centers. Training uses unit weight and calibration-loss checkpoint selection, without downstream labels.

3.2 VALUE CALIBRATION: A SHARED RANGE OBJECTIVE

For any probability vector p, orthogonality and convexity give

$$\left\|pE_V G^\top\right\|_2^2 = \|pE_V\|_2^2 \le \sum_s p_s \left\|e_s^V\right\|_2^2. \tag{8}$$

This bound relates value readout distortion to attention-weighted row errors. We use mean squared range over calibration rows as a tractable proxy, pooling all token/head rows within a layer to learn a shared G:

$$\mathcal{L}_V(G) = \mathbb{E}_v[r_\beta(vG)^2], \qquad r_\beta(x) = \mathrm{softmax}(\beta x)x^\top - \mathrm{softmax}(-\beta x)x^\top, \quad \beta = 24. \tag{9}$$

Hard/smooth K/V combinations differ by less than 1.1% in out-of-calibration readout MSE (Table 11); we use hard K and smooth V. Both rotations use Gaussian QR initialization and Riemannian Adam (K/V learning rates 0.005/0.01; at most 8,000 steps), with calibration-loss checkpoint selection. Appendix F gives full training settings; Appendix D gives approximation and readout bounds.

3.3 KEY-ONLY CENTERING AFTER ROTATION FITTING

Choosing the encoding origin. Rotation preserves mean energy: $\|\mathbb{E}[k]A\|_2^2 = \|\mathbb{E}[k]\|_2^2$. Row range is invariant to a common channel shift, but absolute-value clipping around zero depends on the

encoding origin. Static channelwise centering makes clipping act on deviations from the calibration mean. For $x = kA$ with calibration mean m, residual energy satisfies

$$\mathbb{E}\,\|x - c\|_2^2 = \mathbb{E}\,\|x - m\|_2^2 + \|m - c\|_2^2\,. \tag{10}$$

The mean $c = m$ minimizes this energy and the corresponding moment upper bound on squared row range (Appendix E). With the fitted A fixed, we estimate $\mu = \mathbb{E}_{\mathrm{cal}}[k]$ and store $c_K = \mu A$. For $s \in \mathcal{L}_t$, we encode $z_s = k_s A - c_K$ and compute

$$\widehat{\ell}_{t,a,s} = \frac{q_{t,a} A\big(\mathcal{Q}(z_s) + c_K\big)^\top}{\sqrt{d}} = \ell_{t,a,s} + \frac{q_{t,a} A e_s^\top}{\sqrt{d}}. \tag{11}$$

Each query adds $q_{t,a} A c_K^\top / \sqrt{d}$ to low-bit history scores, recovering the original scores in exact arithmetic without quantization error. Since BF16 keys are not centered, applying this correction only to the low-bit region preserves the regions' relative weights. One center is stored per layer and KV head, with no per-token metadata.

Choosing offsets through attention. The mean-energy identity also applies to values, but K and V have different readout operators. Softmax removes a logit error shared by all visible keys; in a mixed-precision cache, an error shared only by the low-bit history changes the total attention weight assigned to that region. Value errors enter the output directly:

$$\left\| \sum_s p_s f_s \right\|_2^2 = \sum_s p_s^2 \,\|f_s\|_2^2 + \sum_{s \neq r} p_s p_r \langle f_s, f_r \rangle, \tag{12}$$

where f_s is the decoded value error. Equal-row reconstruction averages differ from attention's selective weighting and cross-row interactions. We therefore evaluate K and V offsets separately. Section 4.7 supports key-only centering: it reduces attention error, whereas value centering improves reconstruction but worsens readout. Appendix E gives exact mixed-region bias relations.

3.4 Quantization and Storage

Keys and values use four-level uniform affine INT2 with groups of 128. Before encoding, rows are clipped symmetrically around zero at their 96th absolute-value percentile. Scales and zero points are stored in BF16. The first 64 and most recent 256 tokens remain in BF16; older tokens are converted to low-bit codes as the window advances. The low-bit region uses $2 + (16 + 16)/128 = 2.25$ bits per scalar. For a sequence of length L with n_B BF16 tokens, the average representation cost is

$$b(L) = 2.25 + 13.75 n_B / L. \tag{13}$$

At $L = 65{,}536$ and $n_B = 320$, this gives 2.3171 bits per scalar, or $6.91\times$ compression relative to BF16. Static rotations and centers cost $(H_{\mathrm{KV}} + 1)d^2 + H_{\mathrm{KV}} d$ scalars per layer, independent of sequence length, and are stored separately. Appendix C specifies the quantizer; Table 8 compares storage formats at 64K.

3.5 Fused Mixed-Precision Attention

Value rotations are foldable into value/output projections; the measured runtime applies them online. RRC-KV fuses key centering into KV writes and inverse value rotation into blockwise output reduction, reducing launches and intermediate reads/writes without changing the cache format. BF16 and INT2 paths share rotated queries; INT2 unpacking, dequantization, and attention are computed on the fly. Section 4.8 reports end-to-end measurements.

4 Experiments

4.1 Setup

We evaluate Qwen3-4B-Thinking-2507, Qwen3-8B, Qwen3-32B (Yang et al., 2025), and GLM-4.7, denoted Q4, Q8, Q32, and GLM. We retain checkpoint weight precision; BF16 refers to the

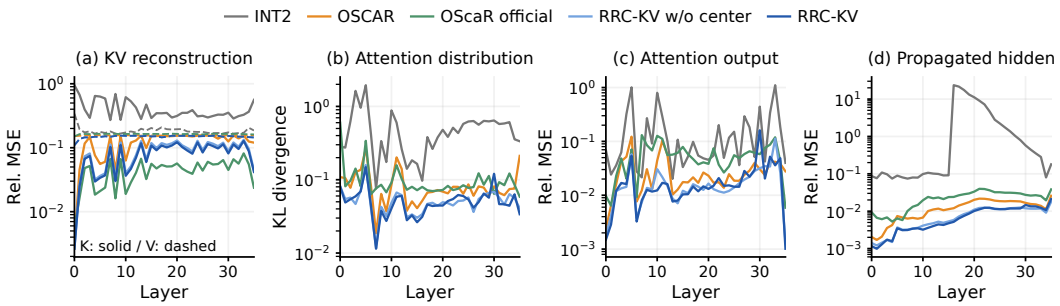


Figure 2: Layerwise Q4 errors, five methods on one 41-token prompt + 1,024-token reference continuation (1,065 total). (a) Original-coordinate KV reconstruction; (b) attention KL; (c) post-W_O output; (d) propagated hidden state. All except KL are relative MSE. Settings and references: Appendix G.

KV cache. OSCAR's public calibration pipeline uses prompts from GPQA Diamond questions (Zhou et al., 2026). To exclude evaluation content from calibration, our main RRC-KV–OSCAR comparison uses the same continuous prefixes of *Alice's Adventures in Wonderland* and RFC 9110, without evaluation questions or chat templates (Carroll, 1865; Fielding et al., 2022). Each text starts independently at position 0; captured QKV activations are pooled to fit rotations and centers. Appendix F gives extraction rules and Q4 token counts.

Baselines. Main accuracy evaluations use official baseline repositories on supported models. OSCAR (Zhou et al., 2026) uses spectral rotations and default channel groups of 128; OScaR official (Su et al., 2026) uses rotation, normalization, and default KIVI-style groups of 32 (K across tokens, V across channels). OSCAR uses Alice/RFC calibration, affine INT2, BF16 metadata, and BF16 prefix/recent windows. OScaR's Transformers path requires batch 1 without chunked prefill; we report five-task accuracy only on supported Qwen models, excluding GLM, RULER, and batched throughput. Its 64K cost is 3.016–3.066 bits per scalar including metadata, norms, and residual tokens (Table 8). TurboQuant NoMP uses three-bit codes without mixed-precision windows (Zandieh et al., 2025). Settings appear in Appendix F.

We choose OScaR as the asymmetric-grouping baseline because it outperforms KIVI, the KV component of QuaRot, and RotateKV in its INT2 LongBench-E comparison (Liu et al., 2024b; Ashkboos et al., 2024; Su et al., 2025; 2026). Published KIVI/QuaRot results appear in Appendix J. Direct INT2 and random orthogonal/Hadamard controls appear in Figure 2 and Table 4; rotation ablations share the quantizer and BF16 windows, with key centering unless disabled explicitly.

Evaluation. We evaluate AIME 2025, GPQA Diamond, MATH-500, HumanEval, and all 1,055 LiveCodeBench v6 problems (Rein et al., 2023; Hendrycks et al., 2021; Chen et al., 2021; Jain et al., 2024). The main table reports means and sample SDs over three generation runs with fixed fitted transforms and a 32K output budget. Coding tasks use temperature 0.2 and one answer per problem for pass@1. Q4/Q8 RULER Full13 covers 8K, 16K, 32K, and 64K, with 650 examples per length, greedy generation, and a 4K output budget (Hsieh et al., 2024). Q8-64K fitting and evaluation both use YaRN factor 4 and RoPE base 10^6; other settings use native RoPE. Full settings are in Appendix F.

4.2 From KV Reconstruction to Attention Propagation

Figure 2 traces quantization errors through all 36 Q4 layers on one fixed BF16 continuation. Panels (a)–(c) fix each layer's reference input to measure local errors; panel (d) propagates each method's hidden states through preceding layers.

Compared with OSCAR under the same quantization configuration, RRC-KV has lower K reconstruction error in all 36 layers, attention KL in 35, and post-W_O output error in 30. Its propagated hidden-state error is lower in every layer, while direct INT2 accumulates large errors in middle and later layers.

Table 1: Five-task accuracy (%), mean ± sample SD over three generation runs per fixed fit. Avg. equally weights tasks; bold marks each column's highest low-bit mean. BPE_{64K} includes metadata and BF16 windows at 64K, not benchmark-specific memory; $*$: OScaR spans 3.02–3.07 with block residuals (Table 8). OSCAR uses Alice/RFC; †: GPQA comparison in Table 3.

Model	Method	AIME25	GPQA	MATH-500	HumanEval	LCB1055	Avg.	BPE_{64K}
Q4	BF16	72.22±5.09	65.32±1.05	96.60±0.20	95.73±0.61	74.63±0.71	80.90	16.00
	RRC-KV	**67.78±5.09**	**67.00±1.91**	**97.00±0.53**	**96.54±1.27**	**71.85±0.57**	**80.03**	2.32
	OSCAR	58.89±5.09	61.79±2.78	96.33±0.46	94.31±1.54	67.68±0.50	75.80	2.32
	OScaR official	65.56±1.93	61.45±2.04	94.00±1.04	94.10±0.93	69.57±0.81	76.94	3.02*
	TurboQuant	18.89±1.92	49.83±3.04	74.13±1.30	9.15±1.61	1.93±0.48	30.79	3.19
Q8	BF16	58.89±6.94	59.26±1.91	96.73±0.12	89.63±1.22	69.76±1.30	74.86	16.00
	RRC-KV	**66.67±6.67**	**57.91±2.59**	95.53±0.76	**89.84±0.93**	**67.96±1.15**	**75.58**	2.32
	OSCAR	62.22±7.70	54.38±1.05	**96.07±0.76**	85.17±0.35	51.00±0.50	69.76	2.32
	OScaR official	62.22±1.92	52.70±1.78	95.73±0.81	80.90±4.66	55.20±0.48	69.35	3.02*
	TurboQuant	41.11±3.85	52.36±2.60	91.80±0.72	69.92±0.93	28.91±0.41	56.82	3.19
Q32	BF16	66.67±6.67	61.28±2.04	95.93±0.46	95.12±1.61	77.85±0.85	79.37	16.00
	RRC-KV	66.67±6.67	61.62±1.75	**95.53±0.23**	**94.51±1.61**	**75.64±0.96**	**78.79**	2.32
	OSCAR†	67.78±3.85	**62.63±1.34**	91.87±1.21	62.40±2.75	37.41±2.36	64.42	2.32
	OScaR official	66.67±3.33	**62.63±2.63**	95.00±1.31	91.66±0.70	74.31±0.28	78.05	3.02*
	TurboQuant	**72.22±5.09**	61.96±2.92	93.60±0.53	93.70±0.93	59.43±1.53	76.18	3.19
GLM	BF16	82.22±1.92	79.63±2.78	96.87±0.58	94.71±0.70	71.66±0.47	85.02	16.00
	RRC-KV	**82.22±5.09**	**77.27±0.51**	96.40±0.20	**94.51±2.20**	68.25±0.34	**83.73**	2.32
	OSCAR	81.11±3.85	76.26±0.51	**96.73±0.50**	92.88±0.93	68.88±1.07	83.17	2.32
	TurboQuant	80.00±6.67	**77.27±1.82**	96.20±0.53	93.49±1.54	**69.23±1.00**	83.24	3.19

Table 2: Q4, two held-out 8K *Pride and Prejudice* passages, all 36 layers: 7,168 BF16 prefill + 1,024 teacher-forced tokens each. K/local post-W_O: pooled relative MSE; KL: query/head average. Lower is better. Full metrics/protocol: Appendix G.

Method	K reconstruction	Attention KL	Local output
INT2	0.701883	1.357490	0.242603
OSCAR	0.056050	0.250272	0.081780
OScaR official	**0.022930**	0.170965	0.049974
RRC-KV w/o center	0.034440	0.136991	0.021066
RRC-KV	0.031842	**0.128706**	**0.020754**

OScaR official reconstructs K better in 35 layers, yet RRC-KV has lower attention-output error in 34 and hidden-state error in all 36. Two held-out 8K passages show the same distinction (Table 2): better K reconstruction does not imply lower attention error. RRC-KV also has lower V reconstruction error (full metrics: Table 10).

4.3 Reasoning and Coding Accuracy

RRC-KV achieves a macro-average of 79.54 versus BF16's 80.04, with the closest five-task mean among low-bit methods on each model (Table 1). Under common Alice/RFC calibration, it exceeds OSCAR on all four models by 6.25 points on average. Q32's coding scores are particularly calibration-sensitive (Table 3); excluding Q32 still gives 79.78 versus 76.24, a 3.54-point gain. Averages and differences use unrounded scores.

Our five-task means also exceed OScaR official on all three Qwen models, while using about 24% fewer KV representation bits at 64K (Table 8); their Q32 scores are close at 78.79/78.05. RRC-KV remains below BF16 on LCB1055 across models by 1.80–3.41 points. Q8's slightly higher overall mean than BF16 (75.58/74.86) should be read in light of finite test sets and generation variability.

TurboQuant performs well on Q32 and GLM but degrades substantially on Q4/Q8, consistent with the model-size trend reported by Zhou et al. (2026). OSCAR remains close to BF16 on GLM, at 83.17 versus 85.02.

Table 3: Calibration configurations (%). Top: HumanEval / LCB1055 on 164 / 1,055 problems, with one fit per configuration and generation seed 1. Bottom: additional Q32 chat-templated Alice/RFC calibration on the same benchmarks.

Model	Method	Alice/RFC	GPQA (8K)
Q4	OSCAR	95.73 / 68.06	93.29 / 68.53
	RRC-KV	95.12 / 71.85	97.56 / 73.18
Q8	OSCAR	84.76 / 51.18	82.32 / 59.91
	RRC-KV	90.85 / 67.20	87.80 / 65.40
Q32	OSCAR	62.20 / 35.64	96.34 / 70.33
	RRC-KV	95.12 / 75.83	93.90 / 75.55

Q32: chat-templated Alice/RFC

Method	HumanEval	LCB1055
OSCAR	58.54	34.98
RRC-KV	93.29	76.40

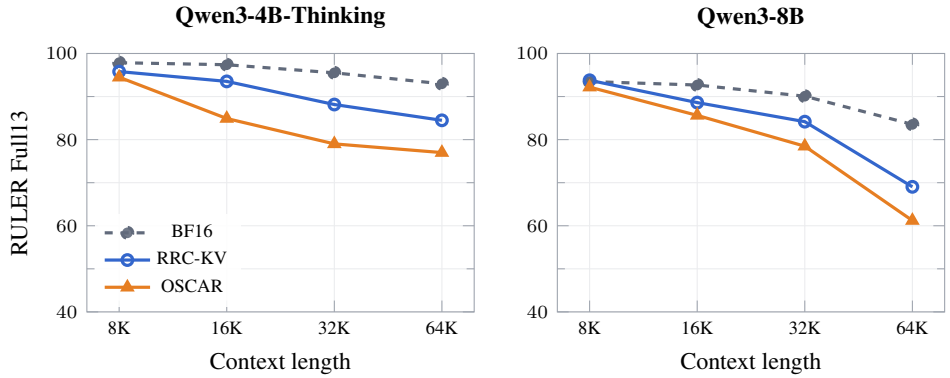


Figure 3: RULER Full13 accuracy across context lengths for BF16, RRC-KV, and OSCAR. Full length-wise results appear in Table 14.

4.4 Calibration Configurations

OSCAR's official repository shows a pronounced coding-accuracy gap from BF16 under Alice/RFC calibration on Q32, unlike other models. On Q4/Q8/Q32, we compare two independent Alice/RFC sequences with an 8K GPQA prompt stream.[1] Each configuration is fitted separately and evaluated on HumanEval and LCB1055 (generation seed 1; 32K output budget). Q32 additionally tests chat-templated Alice/RFC to isolate template effects.

Only Q32's OSCAR scores improve on both tasks with GPQA calibration, by 34.14 and 34.69 points. On Q4/Q8, HumanEval decreases while LCB increases. Between Alice/RFC and GPQA, RRC-KV changes by at most 3.05 points, and only 1.22/0.28 on Q32. With GPQA calibration, RRC-KV leads on both tasks for Q4/Q8; Q32 favors OSCAR on HumanEval and RRC-KV on LCB. Applying a chat template to Alice/RFC does not restore OSCAR's GPQA-calibrated performance (Table 3).

4.5 Long-Context Accuracy

RRC-KV exceeds OSCAR in all eight model/length settings (Figure 3), averaging 87.19 versus 81.61. At 64K, Q4/Q8 score 84.48/69.04, gains of 7.49/7.83 points over OSCAR.

At 64K, multi-key retrieval (`niah_multikey_2/3`) and question answering (`qa_1/2`) account for about 94%/76% of the net BF16 gap on Q4/Q8; all three single-needle tasks score 100/98. Q4's `niah_multikey_3` falls from 98 at 8K to 56 at 64K, suggesting that associating similar candidates remains harder than simple retrieval.

[1]The GPQA calibration length follows the default 8K setting in Table 7 of Zhou et al. (2026).

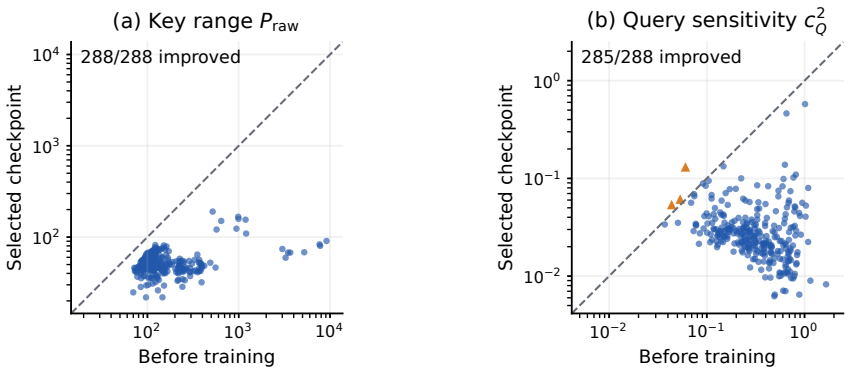


Figure 4: Q4 objectives before/after training at each head's joint-loss-selected checkpoint. All 36 layers $\times$ 8 KV heads are shown; points below $y = x$ improve over random orthogonal initialization.

4.6 Component Ablations: Rotation Learning and Centering

We assess range, sensitivity, and readout offline, and task accuracy on RULER64K.

Optimizing both calibration objectives. Figure 4 shows lower squared range in all 288 key heads and lower sensitivity concentration c_Q^2 in 285, with median after/before ratios of 0.424 and 0.086. Both terms thus improve in 285/288 heads at the jointly selected checkpoints.

What does the query regularizer constrain? Relative to range-only training, the query regularizer lowers c_Q^2 in every head; median paired c_Q^2 and squared-range ratios are 0.056 and 1.001. With matched-initialization weight-sweep assets, r_m decreases in all 288 heads on four out-of-calibration trajectories: its 95th percentile falls from 22.59 to 2.33, while mean squared row range on the calibration data increases by 0.0082%. This supports reduced sensitivity concentration with range reduction largely preserved (Appendix G).

Readout MSE varies by about 3% across the tested weights $\lambda \in \{0, 1, 10\}$; we fix $\lambda = 1$ (Table 11).

Rotation learning and task accuracy. With random or trained V fixed, learning K improves RULER64K by 10.85 or 12.68 points; with trained K, learning V adds 2.78 points (Table 4). Random V also exceeds $V = I$ under trained K, separating the benefit of rotation from that of learning it.

Key-objective ablations fix trained V and K initialization, fitting matching centers. Range-only training gains 10.76 points over random K, whereas query-only training scores 69.36%, identifying range learning as the main source of task gains. Adding the query term gives a 1.92-point higher score in this paired greedy run.

Three K/V training initializations yield 96.00 ± 0.94 on HumanEval and 84.23 ± 1.13 on RULER64K under the same protocol, showing limited variation in this setting (Table 9).

Rotation and key centering. Key centering raises Haar from 0.31 to 70.85% and random-sign Hadamard from 0.05 to 60.46% (Table 4). The latter's uncentered collapse agrees qualitatively with literature-reported QuaRot-style INT2 degradation (Zhou et al., 2026) (Appendix J). RRC-KV exceeds the reported OSCAR scores with the same static centering (84.48 vs. 79.63%) and without (81.02 vs. 76.99%); key centering adds 3.47 points for RRC-KV (paired statistics: Appendix G).

4.7 Readout Effects of K/V Centering

Using the main Q4 fit, CPU diagnostics on four out-of-calibration BF16 trajectories cover all 36 layers. Key centering reduces attention KL by 6.88% and joint post-W_O MSE by 8.33% with uncentered quantized V fixed. Value centering reduces reconstruction MSE by 4.76% but increases fixed-attention post-W_O MSE by 71.77%; readout already worsens before W_O. Leaving the lowest-

Table 4: Q4 RULER64K (%), 650 Full13 examples per run. Lower-panel pairs vary only K centering. Δ vs. RRC-KV uses unrounded scores (centered scores below). Protocol: Appendix G.

Variant	Accuracy	Δ	Variant	Accuracy	Δ
Random K + random V	70.85	−13.63	w/o Query term	82.56	−1.92
Random K + trained V	71.80	−12.68	w/o Range term	69.36	−15.12
Trained K + random V	81.70	−2.78	$V = I$ (trained K)	73.43	—
Fixed K/V rotations			w/o K center	with K center	Δ
Haar random orthogonal			0.31	70.85	−13.63
Random-sign Hadamard (QuaRot-style KV)			0.05	60.46	−24.02
OSCAR spectral K/V			76.99	79.63	—
RRC-KV (trained K/V)			81.02	84.48	—

Table 5: RRC-KV end-to-end throughput change (%), three independent restarts, 1,024 output tokens/request. Q4: TP1; GLM: TP8. All methods use Triton, without FA4.

		16K / batch16		32K / batch8	
Model	GPU	vs. BF16	vs. OSCAR	vs. BF16	vs. OSCAR
Q4	B200	−0.51	+0.99	+0.01	+3.73
Q4	RTX PRO 5000	+50.11	+1.37	+37.84	+6.07
GLM	B200	−5.49	+5.50	−2.09	+5.22

range quartile uncentered removes this aggregate readout increase, linking it to selectively read rows (Appendix H).

4.8 Inference Efficiency

Compression and implementation jointly yield Q4 E2E gains of 37.84%–50.11% over BF16 on RTX PRO 5000, but near parity on B200; GLM TP8 remains slower (Table 5). KV traffic savings compete with unpacking, dequantization, transforms, computation, and communication. E2E exceeds OSCAR in all six workloads despite slightly lower Q4 16K decode rates (Appendix K).

5 Related Work

KurTail/DartQuant learn fusible kurtosis/Whip-based rotations while retaining random Hadamard post-RoPE Q/K transforms (Akhondzadeh et al., 2025; Shao et al., 2025). InfoQuant suppresses row-centered activation peaks via learned rotations (Li et al., 2026); BASE-Q learns post-rotation channel biases (He et al., 2025). We learn per-head post-RoPE rotations on uncentered keys for affine INT2 KV. KV methods use asymmetric layouts and outlier-aware rotations (Liu et al., 2024b; Hooper et al., 2024; Su et al., 2025; 2026); OSCAR uses attention-aware covariance (Zhou et al., 2026). SageAttention centers keys via softmax shift invariance (Zhang et al., 2024). Concurrent OptR[2] learns rotations with static key centers using STE attention-distribution/post-W_O losses (Yun et al., 2026); RRC-KV uses range and query sensitivity without quantizer backpropagation. TurboQuant studies randomly rotated vector quantization and rate–distortion (Zandieh et al., 2025); ours uses affine scalar INT2.

6 Conclusion and Future Work

RRC-KV improves scalar INT2 KV caching through range-driven rotations, query regularization, and static key centering, supported by task evaluations and diagnostics. Future directions include per-head value rotations and readout objectives, position- or length-specific key rotations and centers, cross-segment alignment through causal reading and RoPE fusion, and accuracy–cost trade-offs in clipping, grouping, and cache windows.

[2]OptR was first posted in August 2026, within two months of the submission deadline.

AI USE STATEMENT

We used ChatGPT and Codex to polish English writing and assist with implementing experimental and analysis code, along with supporting literature and implementation checks. The research ideas and methodological design were developed by the authors, without AI involvement in their formulation or refinement. All reported results were obtained through model evaluation and benchmark scoring. The authors are responsible for the paper's content and results.

REPRODUCIBILITY STATEMENT

The appendices provide proofs, exact quantization rules, optimization and generation settings, full Q4/Q8 RULER results, and local-attention diagnostic protocols. Main evaluations use public calibration text without benchmark questions or answers and keep fitted transforms fixed across each task's three generation runs. Table 3 separately compares calibration configurations that include a GPQA prompt stream. Upon acceptance, we will release the implementation, calibration and evaluation scripts, and inference kernels.

REFERENCES

Mohammad Sadegh Akhondzadeh, Aleksandar Bojchevski, Evangelos Eleftheriou, and Martino Dazzi. KurTail: Kurtosis-based LLM quantization. *arXiv preprint arXiv:2503.01483*, 2025. URL https://arxiv.org/abs/2503.01483.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-Free 4-Bit Inference in Rotated LLMs. *arXiv preprint arXiv:2404.00456*, 2024. URL https://arxiv.org/abs/2404.00456.

Lewis Carroll. *Alice's Adventures in Wonderland*. Macmillan, 1865. URL https://www.gutenberg.org/ebooks/11.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Roy T. Fielding, Mark Nottingham, and Julian Reschke. HTTP Semantics. Technical Report RFC 9110, RFC Editor, 2022. URL https://www.rfc-editor.org/rfc/rfc9110.html.

Liulu He, Shenli Zheng, Karwei Sun, Yijiang Liu, Yufei Zhao, Chongkang Tan, Huanrui Yang, Yuan Du, and Li Du. BASE-Q: Bias and asymmetric scaling enhanced rotational quantization for large language models. *arXiv preprint arXiv:2506.15689*, 2025. URL https://arxiv.org/abs/2506.15689.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/abs/2103.03874.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization. *arXiv preprint arXiv:2401.18079*, 2024. URL https://arxiv.org/abs/2401.18079.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the Real Context Size of Your Long-Context Language Models? *arXiv preprint arXiv:2404.06654*, 2024. URL https://arxiv.org/abs/2404.06654.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

Ke Li, Dong An, Xiaoling Zang, Can Ye, Liang Xie, Qibo Qiu, Chen Shen, Xiaofei He, and Wenxiao Wang. InfoQuant: Shaping activation distributions for low-bit LLM quantization. *arXiv preprint arXiv:2605.26175*, 2026. URL https://arxiv.org/abs/2605.26175.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM Quantization with Learned Rotations. *arXiv preprint arXiv:2405.16406*, 2024a. URL https://arxiv.org/abs/2405.16406.

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. *arXiv preprint arXiv:2402.02750*, 2024b. URL https://arxiv.org/abs/2402.02750.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Yuantian Shao, Yuanteng Chen, Peisong Wang, Jianlin Yu, Jing Lin, Yiwu Yao, Zhihui Wei, and Jian Cheng. DartQuant: Efficient rotational distribution calibration for LLM quantization. *arXiv preprint arXiv:2511.04063*, 2025. URL https://arxiv.org/abs/2511.04063.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Zunhai Su, Zhe Chen, Wang Shen, Hanyu Wei, Linge Li, Huangqi Yu, and Kehong Yuan. RotateKV: Accurate and Robust 2-Bit KV Cache Quantization for LLMs via Outlier-Aware Adaptive Rotations. *arXiv preprint arXiv:2501.16383*, 2025. URL https://arxiv.org/abs/2501.16383.

Zunhai Su, Rui Yang, Chao Zhang, Yaxiu Liu, Yifan Zhang, Wei Wu, Jing Xiong, Dayou Du, Xialie Zhuang, Yulei Qian, Yuchen Xie, Yik-Chung Wu, Hongxia Yang, and Ngai Wong. OScaR: The occam's razor for extreme KV cache quantization in LLMs and beyond, 2026. URL https://arxiv.org/abs/2605.19660.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. *arXiv preprint arXiv:2211.10438*, 2022. URL https://arxiv.org/abs/2211.10438.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Vincent-Daniel Yun, Woosang Lim, Minsoo Cheong, Sunwoo Lee, Murali Annavaram, Sai Praneeth Karimireddy, and Sungjoo Yoo. Output-Aware Rotation for INT2 KV-Cache Quantization. *arXiv preprint arXiv:2608.02691*, 2026. URL https://arxiv.org/abs/2608.02691.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online Vector Quantization with Near-optimal Distortion Rate. *arXiv preprint arXiv:2504.19874*, 2025. URL https://arxiv.org/abs/2504.19874.

Jintao Zhang, Jia Wei, Haofeng Huang, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-Bit Attention for Plug-and-play Inference Acceleration. *arXiv preprint arXiv:2410.02367*, 2024. URL https://arxiv.org/abs/2410.02367.

Zhongzhu Zhou, Donglin Zhuang, Jisen Li, Ziyan Chen, Shuaiwen Leon Song, Ben Athiwaratkun, and Xiaoxia Wu. OSCAR: Offline Spectral Covariance-Aware Rotation for 2-bit KV Cache Quantization. *arXiv preprint arXiv:2605.17757*, 2026. URL https://arxiv.org/abs/2605.17757.

A Exact Identities and Causal Pairing

The following identities use unquantized attention as the reference and hold in exact arithmetic. For orthogonal matrices A and G,

$$(qA)(kA)^\top = qk^\top, \qquad (pVG)G^\top = pV.$$

For any fixed center c_K, writing $\mathcal{Q}(kA - c_K) + c_K = kA + e^K$ gives the key-error relation in Eq. (2). Expanding $\widehat{p}(V + E_V G^\top) - pV$ gives the output-error decomposition. Encoding error in the BF16 region is defined as zero relative to the unquantized reference path; finite-precision transform operations introduce a separate numerical error.

For a stored key row s, square $\delta\ell_{t,a,s} = q_{t,a}A(e_s^K)^\top/\sqrt{d}$ and sum over all reading positions satisfying $s \in \mathcal{L}_t$. Regrouping by s gives

$$\sum_{t,a}\sum_{s\in\mathcal{L}_t}\delta\ell_{t,a,s}^2 = \frac{1}{d}\sum_s e_s^K A^\top C_Q^{\mathcal{L}}(s) A (e_s^K)^\top, \quad C_Q^{\mathcal{L}}(s) = \sum_{\substack{a\in\mathcal{H}(h)\\ t:\, s\in\mathcal{L}_t}} q_{t,a}^\top q_{t,a}. \tag{14}$$

If a row is re-encoded at runtime, its error can be indexed by both t and s.

A.1 From Causal Pairing to a Product-of-Marginals Proxy

Sample (q, e) from the distribution of actual visible low-bit query–key pairs, and define $X = A^\top q^\top qA$, $Y = e^\top e$, $M_* = \mathbb{E}X$, and $C_* = \mathbb{E}Y$. Then

$$\mathcal{E}_{\text{pair}} = \frac{1}{d}\operatorname{tr}(M_*C_*) + \frac{1}{d}\mathbb{E}\operatorname{tr}[(X - M_*)(Y - C_*)]. \tag{15}$$

This follows by expanding the centered matrices. The sufficient condition $\mathbb{E}[X \mid Y] = M_*$ makes the correlation term zero. The calibration query moment $M = A^\top C_Q A$ and the uniformly weighted row-error moment C_E may also differ from the causal-pairing marginals. Their effect can be decomposed as

$$\operatorname{tr}(M_*C_*) - \operatorname{tr}(MC_E) = \operatorname{tr}((M_* - M)C_*) + \operatorname{tr}(M(C_* - C_E)).$$

These identities describe the effects of correlation, query reweighting, and key-row reweighting. Equation (3) is the exact risk under the specified product of marginal distributions.

B Error-Allocation Motivation for the Query Regularizer

Proposition 1 (Sensitivity to coordinate-wise error allocation)**.** *Let $M, C_E \succeq 0$, $m = \operatorname{diag} M$, $w = \operatorname{diag} C_E$, $\bar{m} > 0$, and $T_E = \operatorname{tr} C_E > 0$. Define c_Q as in Eq.* (6) *and $\rho_E = \|w - (T_E/d)\mathbf{1}\|_2 / T_E$. Then*

$$\frac{1}{d}\operatorname{tr}(MC_E) \le \frac{\bar{m}T_E}{d}\left(1 + \sqrt{d-1}\rho_E c_Q\right) + \frac{\|\operatorname{off} M\|_F \, \|\operatorname{off} C_E\|_F}{d}. \tag{16}$$

Proof of Proposition 1. Let $m_c = m - \bar{m}\mathbf{1}$ and $w_c = w - (T_E/d)\mathbf{1}$. Both have zero coordinate sum, so

$$m^\top w = \bar{m}T_E + m_c^\top w_c \le \bar{m}T_E + \|m_c\|_2 \, \|w_c\|_2 = \bar{m}T_E(1 + \sqrt{d-1}\rho_E c_Q).$$

For symmetric matrices, $\operatorname{tr}(MC_E) = m^\top w + \langle \operatorname{off} M, \operatorname{off} C_E\rangle_F$. Apply the Frobenius form of the Cauchy–Schwarz inequality to the second term and divide by d. □

Worst-case diagonal allocation. If only the diagonal error budget is constrained, then

$$\sup_{\substack{C_E\succeq 0,\ C_E \text{ diagonal}\\ \operatorname{tr} C_E = T_E}} \frac{1}{d}\operatorname{tr}(MC_E) = \frac{T_E}{d}\max_j m_j, \qquad \max_j m_j \le \bar{m}\left(1 + \frac{d-1}{\sqrt{d}}c_Q\right).$$

The equality assigns all error to the most sensitive coordinate. Relative to the uniform diagonal allocation $\bar{m}T_E/d$, the worst-case amplification factor is exactly $r_m = \max_j m_j/\bar{m}$. For the latter bound, let the largest centered coordinate be a. The remaining coordinates sum to $-a$, so their squared norm is at least $a^2/(d-1)$. Thus, the diagonal CV bounds worst-case allocation amplification associated with coordinate sensitivity concentration. If arbitrary PSD error covariances are allowed, the corresponding supremum is $T_E\lambda_{\max}(M)/d$; Proposition 1 explicitly retains the off-diagonal contribution.

An additive proxy bound. Suppose $T_E \le \gamma P$, $0 \le P \le P_{\max}$, and $\rho_E \le \rho_0$ on the distribution being analyzed. Orthogonality fixes $\bar{m} = \operatorname{tr}(C_Q)/d$. Let $k_0 = \sqrt{d-1}\rho_0$ and collect the off-diagonal and pairing terms in Δ. Proposition 1 and $2c_Q \le 1 + c_Q^2$ give

$$\mathcal{E} \le \frac{\bar{m}\gamma}{d}\left[\left(1 + \frac{k_0}{2}\right)P + \frac{k_0 P_{\max}}{2}c_Q^2\right] + \Delta. \tag{17}$$

Bounded additional quantization error can also be included in Δ. For finitely many calibration rows, compactness of the orthogonal group and continuity of the row range ensure that $P_{\max}$ exists. Appendix E relates raw and centered key ranges. This bound separates the effects of range and sensitivity on the proxy risk and motivates an additive objective; the unit weight used in practice is supported by the offline ablation in Table 11.

C Uniform Affine INT2 Quantization

For a row vector x with $d = 128$, first compute in FP32. Sort the absolute coordinate values and select zero-based index $\min(d-1, \lfloor 0.96d \rfloor) = 122$, giving the 123rd order statistic τ. Let $x_c = \operatorname{clip}(x, -\tau, \tau)$, $a = \min_j x_{c,j}$, and $b = \max_j x_{c,j}$, and compute

$$s = \max(b - a, 10^{-8})/3, \qquad z_0 = -a/s, \tag{18}$$

$$c_j = \operatorname{clip}(\lfloor x_{c,j}/s + z_0 + 0.5 \rfloor, 0, 3), \tag{19}$$

$$\widetilde{s} = \mathrm{BF16}(s), \qquad \widetilde{z}_0 = \mathrm{BF16}(z_0), \tag{20}$$

$$\widehat{x}_j = (c_j - \mathrm{FP32}(\widetilde{z}_0))\,\mathrm{FP32}(\widetilde{s}). \tag{21}$$

The zero point is dimensionless. Constant rows also use the scale floor. Codes are determined before metadata is rounded to BF16.

Centering and the clipping origin. For a fixed channelwise center c, clipping in residual coordinates and restoring the center gives

$$T_c(x) = c + \operatorname{clip}(x - c, -\tau_c, \tau_c), \qquad \tau_c = \mathrm{Percentile}_{0.96}(|x - c|),$$

using the order statistic specified above. A common shift $b\mathbf{1}$ leaves $\operatorname{rng}(x)$ unchanged but changes absolute magnitudes and potentially the clipped coordinates. Thus, centering changes the clipping reference as well as the affine grid. The arithmetic above isolates BF16 metadata rounding; Appendix H additionally includes BF16 rounding after center subtraction and in the two decoding operations.

Error budget. Let $\widehat{x}_{32} = (c - z_0)s$ be the result of decoding the same codes with FP32 metadata. Then

$$\widehat{x} - x = (x_c - x) + (\widehat{x}_{32} - x_c) + (\widehat{x} - \widehat{x}_{32}) = e_{\mathrm{clip}} + e_{\mathrm{round}} + e_{\mathrm{meta}}.$$

Ignoring arithmetic error in grid selection, nearest-grid rounding satisfies $\|e_{\mathrm{round}}\|_2 \le \sqrt{d}\, s/2$. With $\delta s = \widetilde{s} - s$ and $\delta z = \widetilde{z}_0 - z_0$, we have

$$e_{\mathrm{meta},j} = (c_j - z_0)\delta s - s\delta z - \delta s \delta z, \qquad \|\widehat{x} - x\|_2 \le \|e_{\mathrm{clip}}\|_2 + \sqrt{d}\, s/2 + \|e_{\mathrm{meta}}\|_2 .$$

Clipping cannot increase the range, so the grid term is controlled by the raw row range, apart from the scale floor. The full budget also includes clipping and metadata errors. These terms can interact; the decomposition is not a sum of independent error variances.

D Value Range and Readout Bounds

Let $f_s = e_s^V G^\top$. Convexity of the squared norm gives

$$\left\| \sum_s p_s f_s \right\|_2^2 \le \sum_s p_s \|f_s\|_2^2 = \sum_s p_s \left\| e_s^V \right\|_2^2 .$$

If the attention-selected row distribution ν_{att} has density ratio $d\nu_{\mathrm{att}}/d\nu \le \kappa$ relative to the reference row distribution ν, taking expectations gives

$$\mathbb{E}\,\|\Delta o_V\|_2^2 \le \kappa \mathbb{E}_{v\sim\nu} \left\| e^V \right\|_2^2 .$$

This condition makes explicit how control of uniform row risk translates into control of attention-weighted readout error.

Smooth range. For $w_+ = \mathrm{softmax}(\beta x)$, use the entropy identity

$$\log \sum_j e^{\beta x_j} = \beta w_+ x^\top + H(w_+)$$

and $H(w_+) \le \log d$ to obtain $\max x - w_+ x^\top \le \log d/\beta$. Apply the same argument to $-x$. Since the derivative of $\mathrm{softmax}(\beta x)x^\top$ with respect to β is a nonnegative variance,

$$0 \le r_\beta(x) \le \mathrm{rng}(x), \qquad \mathrm{rng}(x) - r_\beta(x) \le 2\log d/\beta.$$

The L_2 triangle inequality further gives

$$\mathbb{E}[\mathrm{rng}(vG)^2] \le \left(\sqrt{\mathcal{L}_V(G)} + \frac{2\log d}{\beta}\right)^2.$$

A shared coordinate basis. Let $L_h(G)$ be the population risk of head h and $\bar{L}(G)$ its average across heads. Suppose $\sup_{h,G} |L_h(G) - \bar{L}(G)| \le b$ and the pooled empirical risk differs uniformly from $\bar{L}$ by at most ϵ. An empirical solution with optimality gap η satisfies

$$L_h(\widehat{G}) \le \inf_G L_h(G) + 2b + 2\epsilon + \eta.$$

This follows by applying both the head-to-average deviation bound and the empirical-risk deviation bound twice. It describes the trade-off between head heterogeneity and shared estimation. We use a layer-shared value rotation; this deterministic deviation result does not require independence between neighboring tokens or heads.

E Static Translation and Two-Stage Calibration

E.1 The Mean as an Encoding Origin

Let $x = kA$ have finite second moments and mean m. For any fixed positive semidefinite matrix W, expansion around the mean gives

$$\mathbb{E}[(x-c)W(x-c)^\top] = \mathbb{E}[(x-m)W(x-m)^\top] + (m-c)W(m-c)^\top. \tag{22}$$

The cross terms vanish because $\mathbb{E}[x-m] = 0$. The mean therefore minimizes all such fixed quadratic residual-energy proxies; the minimizer may be nonunique when W is singular. Setting $W = I$ gives Eq. (10). Let $H = I - \mathbf{1}\mathbf{1}^\top/d$. Every channel difference is the inner product of yH with a vector of norm $\sqrt{2}$, so Cauchy–Schwarz gives $\mathrm{rng}(y)^2 \le 2\,\|yH\|_2^2$. Hence

$$\mathbb{E}[\mathrm{rng}(x-c)^2] \le 2\mathbb{E}[\|(x-c)H\|_2^2],$$

The right-hand side is minimized by $c = m$, giving a moment-bound justification for choosing the mean as the encoding origin. Clipping, grid rounding, and BF16 metadata jointly determine the actual reconstruction residual; Appendix C gives its error decomposition.

E.2 Range Control across Calibration Stages

Proposition 2 (Range risk under bounded drift along the mean direction)**.** *Let ξ have finite second moments and satisfy $\xi \overset{d}{=} -\xi$. For a fixed vector μ and matrix A, define $F_A(\eta) = \mathbb{E}[\mathrm{rng}((\xi + \eta\mu)A)^2]$. Then F_A is even and convex, and*

$$F_A(0) \le F_A(1) = \sup_{|\eta|\le 1} F_A(\eta).$$

Proof. The range is the nonnegative convex function $\max_{i,j}(x_i - x_j)$, so its square is also convex. Affine composition and expectation preserve convexity. The range is even, which, together with symmetry of ξ, gives $F_A(-\eta) = F_A(\eta)$. A convex function on a closed interval attains a maximum at an endpoint, and the two endpoint values are equal. □

For raw calibration keys $k = \mu + \xi$, $P_{\mathrm{raw}}(A) = F_A(1)$ and the centered range risk is $F_A(0)$. Adding a query regularizer independent of η preserves the pointwise worst-case identity. This interpretation assumes centrally symmetric residuals of fixed shape, with drift along the calibration mean direction.

A distribution-free range relation. Let $\mu = \mathbb{E}k$. Jensen's inequality gives $\mathrm{rng}(\mu A)^2 \le P_{\mathrm{raw}}(A)$. Range subadditivity and the L_2 triangle inequality yield

$$\sqrt{\mathbb{E}[\mathrm{rng}((k-\mu)A)^2]} \le \sqrt{P_{\mathrm{raw}}(A)} + \mathrm{rng}(\mu A) \le 2\sqrt{P_{\mathrm{raw}}(A)},$$

and thus

$$\mathbb{E}[\mathrm{rng}((k-\mu)A)^2] \le 4P_{\mathrm{raw}}(A). \tag{23}$$

The raw-key range objective therefore controls an upper bound on centered range risk, while mean estimation determines the encoding origin. For an additional shift δ,

$$\sqrt{\mathbb{E}[\mathrm{rng}((k-\mu+\delta)A)^2]} \le \sqrt{\mathbb{E}[\mathrm{rng}((k-\mu)A)^2]} + \mathrm{rng}(\delta A).$$

These results concern range risk. Corresponding quantization-error bounds follow when the clipping and metadata budgets in Appendix C are controlled.

E.3 Mixed-Precision Attention and Region Bias

For a row probability vector p, define $J_p = \mathrm{Diag}(p) - p^\top p$. A small logit perturbation u gives the first-order approximation

$$\delta p \approx u J_p, \qquad \delta o_K \approx \sum_s p_s u_s (v_s - o).$$

A common component $u = c\mathbf{1}$ cancels, but a shift restricted to the low-bit history region does not. Specifically, for $u_s = c\mathbf{1}\{s \in \mathcal{L}\}$ and $P_{\mathcal{L}} = \sum_{s\in\mathcal{L}} p_s$,

$$\widehat{P}_{\mathcal{L}} = \frac{e^c P_{\mathcal{L}}}{1 - P_{\mathcal{L}} + e^c P_{\mathcal{L}}}, \qquad (\widehat{p} - p)V = (\widehat{P}_{\mathcal{L}} - P_{\mathcal{L}})(\bar{v}_{\mathcal{L}} - \bar{v}_{\mathcal{B}}).$$

When both regions have positive probability mass, $\bar{v}_{\mathcal{L}}$ and $\bar{v}_{\mathcal{B}}$ are their conditional weighted means. Relative probabilities within each region remain unchanged. The center correction must therefore be restored to exactly the region that was encoded.

F Implementation and Evaluation Settings

Table 6: Calibration and representation settings.

Setting	Value
Calibration sources	Alice and RFC 9110, continuous sequences
K granularity / space	Per layer and KV head / post-RoPE
K objective	Squared hard range plus owning-query CV^2
Query variance convention	Sample variance across coordinates, denominator $d-1$
V granularity / objective	Shared per layer / squared smooth range, $\beta = 24$
Initialization / optimizer	Gaussian QR / Riemannian Adam
K / V learning rate	0.005 / 0.01
Maximum steps / check interval	8,000 / 100
Stopping	Less than 1% improvement in two 400-step windows
Checkpoint selection	Lowest calibration objective
Center	Joint-text mean, fitted after key rotation, K only
Codec	Uniform affine INT2, group 128, clip ratio 0.96
Metadata	BF16 scale and dimensionless zero point
BF16 window / flush granularity	Prefix 64, recent 256 / 8 tokens

Table 6 lists our default calibration and representation settings. QuaRot uses a fixed KV clipping ratio of 0.95, and OSCAR specifies a default key clipping ratio of 0.96 (Ashkboos et al., 2024; Zhou et al., 2026). We use 0.96 for both K and V by default; Appendix C specifies the percentile rule. Value optimization backpropagates $\mathcal{L}_V / \mathrm{stopgrad}(\max(\mathcal{L}_V, 10^{-12}))$ to adjust the gradient scale, while stopping and checkpoint selection use the original $\mathcal{L}_V$. Keys directly optimize Eq. (7). Rotations are fitted on raw key trajectories, after which the static mean is mapped into the fitted coordinates.

Rotation initialization. Both K and V rotations are initialized from a matrix Z with independent standard Gaussian entries. We factor $Z = QR$, then normalize the column signs of Q using the diagonal of R, obtaining $Q\,\mathrm{diag}(\mathrm{sign}(R_{jj}))$. In exact arithmetic, this construction samples from the Haar distribution on the orthogonal group. The implementation uses FP32, and sign normalization does not depend on calibration activations.

In one Q4 run on a single RTX PRO 5000, our implementation completed K/V fitting and artifact finalization in approximately 40.3 minutes of wall-clock time, including subprocess startup, data I/O, center construction, merging, and validation, but excluding QKV trace collection.

Table 7: Generation and benchmark settings for our evaluation.

Evaluation	Temperature	Top-p	Top-k	Output limit
Qwen reasoning / mathematics	0.6	0.95	20	32,768
GLM reasoning / mathematics	1.0	0.95	Disabled	32,768
HumanEval / LiveCodeBench	0.2	0.95	Disabled	32,768
RULER	0	1	20	4,096

Table 7 summarizes sampling settings. The five benchmarks contain 30 AIME 2025, 198 GPQA Diamond, 500 MATH, 164 HumanEval, and 1,055 LiveCodeBench v6 questions. Code generation uses temperature 0.2, following the pass@1 setting in the original HumanEval paper and the default temperature in the official LiveCodeBench implementation, respectively (Chen et al., 2021; Jain et al., 2024). Each run generates one answer per question and reports pass@1 using official tests. MATH uses a fixed version of math-verify. Our five main evaluations use generation seeds 1, 2, and 3. RULER uses greedy generation with client base seed 42; the Q8-64K client derives request seeds from task and example identifiers. RULER averages its 13 tasks equally. Main-table standard deviations describe generation randomness with fixed rotations and centers; averages across tasks or models are equally weighted macro-averages.

Baselines and backends. For each model, our five main evaluations use the same K/V rotations and matching centers, while the OSCAR reproduction uses Alice/RFC spectral calibration. OSCAR's official serving example uses FP32 quantization scale/zero metadata; our controlled reproduction uses BF16, matching RRC-KV's storage accounting. In the Q32 calibration comparison, both OSCAR configurations retain K/V clipping ratios of 0.96/0.92 and BF16 metadata; both RRC-KV configurations use 0.96/0.96 and BF16 metadata. Baseline evaluations in Table 1 and Figure 3 use the official repositories without source-code modifications. RULER uses chunked prefill. Models retain their native reasoning and chat configurations and checkpoint weight precision, including the FP8 checkpoint for GLM-4.7.

Calibration configurations and text processing. Standard calibration uses continuous prefixes of two public texts, ending at paragraph boundaries. For Alice, we remove the Project Gutenberg wrapper and start at the beginning of the document, retaining the title and contents. For RFC 9110, we remove page headers and footers and start at Section 1, Introduction. Neither text uses a chat template. QKV activations are captured in separate requests, each starting at position 0, rather than concatenating the texts into one context. For Q4, the sequences contain 18,957 and 17,611 tokens, totaling 36,568; both complete trajectories are used at every layer to fit rotations and static key centers. Pooling their activations during fitting does not change their separate contexts or position indices. These exact token counts are specific to the Q4 tokenizer.

The upper panel of Table 3 fits each calibration configuration separately and fixes generation seed 1; these single-run results are separate from the three-run means in the main table. Alice/RFC uses the independent text sequences described above, whereas GPQA uses one continuous 8,192-token prompt sequence. In the original Q32 two-configuration comparison, the public texts have no chat template, while GPQA question–answer prompts are rendered with the model's chat template. This comparison changes the full configuration, including content, format, and length. The lower panel adds Q32 HumanEval and LCB1055 results for chat-templated Alice/RFC.

Storage accounting at a common length. Table 8 fixes $L = 65{,}536$, $d = 128$, and $H_{\mathrm{KV}} = 8$, averaging codes, per-group/per-token metadata, and retained unquantized entries over all K/V

scalars. The BPE column in Table 1 rounds this accounting to two decimals. GLM also uses $d = 128$; the BF16, RRC-KV, OSCAR, and TurboQuant per-scalar formulas below apply unchanged and do not depend on layer count or TP degree. OScaR official is reported only for Qwen; its main-table BPE is the archived-block value, with the residual-dependent range given in the caption. Model weights, static rotations/codebooks/centers, reserved allocator pools, and temporary kernel tensors are accounted for separately. These are cache-format costs at a common reference length, not actual benchmark lengths or measured runtime memory.

Table 8: KV storage accounting at 64K tokens ($d = 128$, $H_{\mathrm{KV}} = 8$). Bits are averaged over K and V scalars, including token-dependent metadata and retained high-precision entries.

Method	Code bits	Metadata	Retained tokens	Bits/scalar
BF16	16	None	All	16.000
RRC-KV	2	BF16 scale/zero, g128	Prefix64 + recent256	2.317
OSCAR	2	BF16 scale/zero, g128	Prefix64 + recent256	2.317
OScaR official	2	16-bit scale/bias, g32; K norm	Block residual $r < 256$	3.016–3.066
TurboQuant NoMP	3	FP16 K norm; V scale/zero	None	3.188

For RRC-KV and OSCAR, the INT2 region uses $2 + 32/128 = 2.25$ bits per scalar; including 320 BF16 tokens gives Eq. (13). OScaR official groups keys along tokens and values along channels, both with group size 32 and two 16-bit parameters per group, giving a base cost of $2 + 32/32 = 3$. Its native path also stores one FP32 key norm per layer and token, shared across KV heads, adding $32/(2H_{\mathrm{KV}}d)$ bits on average. The residual region in its 256-token block scheme changes as the sequence advances, so we report the range of $3 + 13r/L + 32/(2H_{\mathrm{KV}}d)$; after a full block is archived, $r = 0$ attains the lower bound. TurboQuant stores 3-bit codes and one FP16 norm per K vector, and 3-bit codes plus FP16 scale/zero per V vector, giving $3 + 48/(2d) = 3.1875$ bits per scalar. NoMP retains no BF16 layers or token windows.

G Training Objectives and Error-Propagation Protocols

Random-sign Hadamard control (QuaRot-style KV). For Q4 with BF16 weights, Table 4 replaces both K and V rotations with normalized Hadamard transforms with random signs, fixed without training. This QuaRot-style KV control applies the rotation within our shared inference pipeline (Ashkboos et al., 2024). It reuses the main Alice/RFC trajectories and the elementwise identical raw key mean μ, recomputing the matching center with the same function for this key transform. The quantizer, clipping, metadata, and BF16 windows follow Table 6. Each centering pair reuses its respective K/V rotations and toggles only key centering. When enabled, centering applies only to keys in the INT2 history region, with the center contribution restored in logits; values and BF16 windows are not centered. Evaluation uses all 650 RULER64K Full13 examples.

Combinations of random and trained rotations. Table 4 reports random K + random V, random K + trained V, trained K + random V, and trained K + trained V, with RULER64K scores of 70.85%, 71.80%, 81.70%, and 84.48%, respectively. Random rotations remain fixed without training; trained K/V use the main objectives. The random/random configuration retains the same Triton settings, quantizer, BF16 windows, and key-only centering as the Hadamard control. The 73.43% result with trained K and $V = I$ is an additional reference, distinct from random orthogonal V.

Runtime checks for the no-centering controls. The Haar no-centering control reuses the same random K/V rotations and disables only key centering. We checked production KV encoding, prefill, and decode against independent reference computations, including attention checks on a fixed 65,234-token RULER input. The checks passed their prespecified numerical criteria. For Haar, enabling the centering path with a zero center reproduced the no-centering generation on this input.

OSCAR rotations with static key centering. The OSCAR row in Table 4 retains OSCAR's spectral K/V rotations without retraining and applies the same static key-centering procedure as RRC-KV: the original calibration Key mean μ is transformed by OSCAR's own key rotation, $c_{\mathrm{OSCAR}} = \mu A_{\mathrm{OSCAR}}$. Values remain uncentered. With K/V rotations and all other evaluation

settings fixed, enabling only key centering raises RULER64K from 76.99% to 79.63% on the same 650 Full13 examples.

Training-objective diagnostics. Figure 4 uses the same rotations as the full key-training configuration in Table 4, covering all 36 Q4 layers and eight KV heads per layer. Initial values are step-0 records from random orthogonal initialization. For each head, we select the first recorded checkpoint attaining its minimum joint training loss. Both objective terms for that head come from this checkpoint and are training-audit values before the orthogonal projection used for export. The range term decreases in all 288 heads and the query term in 285; the three increases in the latter are also shown.

Range-only training and query regularization. The objective comparison in Section 4.6 uses untrained random rotations, range-only training, and full additive-objective training with the same initialization and calibration data, covering the same 288 layer/head pairs. Each training configuration selects the first minimum of its own calibration loss. We measure P_{raw} and c_Q^2 after the export-time orthogonal projection, compute paired ratios per head, and report their medians. Relative to random rotations, the median ratios for the two terms are 0.420 and 1.604 for range-only training, and 0.424 and 0.086 for full training. Both reduce range in every head. Full training gives lower c_Q^2 than random rotations in 285 heads and lower c_Q^2 than range-only training in every head. These statistics describe the calibration objectives; Table 4 separately reports task accuracy.

In Table 4, the full objective has a paired mean-score gain of $+1.923$ points over removing query regularization. The task-stratified paired bootstrap 95% confidence interval is $[-0.715, 4.562]$ points, and the two-sided paired permutation test gives $p = 0.1642$. The analysis preserves partial credit and uses 650 matched examples and 100,000 bootstrap resamples.

Paired effect of key centering. With the original main-evaluation K/V assets fixed and V uncentered, key centering raises RULER64K from 81.02% to 84.48%, a $+3.47$-point gain computed from unrounded scores. All 650 examples match in prompt, reference, output budget, and fitted assets. Preserving partial credit, we resample the 50 paired examples within each of 13 tasks and average task means. With 100,000 resamples, the percentile bootstrap 95% interval is $[1.06, 5.88]$ points; 100,000 independent within-pair sign swaps give a two-sided $p = 0.00518$. PCG64 seeds are 20260926 and 20260927, respectively, with a linear-interpolation percentile and add-one permutation correction. This measures example-level variation for the fixed fit. The `niah_multikey_2/3` scores rise from 68/44 to 80/56 (+12 each).

Out-of-calibration coordinate sensitivity. We use the matched-initialization $\lambda = 0$ and 1 rotations from the weight sweep in Table 11 and its four fixed out-of-calibration trajectories, totaling 65,536 tokens. For each KV head, we pool second moments over all associated query heads and tokens, weighted by sample count, then compute $M(A) = A^\top C_Q A$ and r_m. Centers are not refitted on these texts. All 288 layer/head pairs have lower r_m: the median falls from 5.88 to 1.62, the 95th percentile from 22.59 to 2.33, and the maximum from 58.76 to 4.73. Each trajectory separately also shows decreases in all 288 heads; percentiles describe variation across heads. Mean squared row range on the calibration data changes from 54.82365 to 54.82815, an increase of 0.0082%. These measurements concern sensitivity concentration across coordinates; orthogonal rotation preserves total query second-moment energy.

K/V training initialization. Table 9 compares three Q4 training seeds. Seed 1235 is the main configuration; seeds 1234 and 1236 each refit both K and V under the same calibration and evaluation protocol. HumanEval is first averaged over three generation seeds per fit, while RULER64K uses one greedy evaluation on 650 Full13 examples. The summary gives the mean and sample standard deviation across the three fits, unlike the generation standard deviations for a fixed fit in Table 1.

Layerwise diagnostics on a fixed sequence. Figure 2 uses the first AIME25-I question. The BF16 model greedily generates a 1,024-token reference continuation, forming a fixed 1,065-token sequence with the 41-token prompt. All five methods replay the same token IDs through all 36 layers. RRC-KV uses the component-experiment rotations and matching centers from Figure 4; the uncentered variant changes only the centering switch. OSCAR uses spectral rotations constructed from the same QKV calibration data. This analysis measures representation error on a fixed sequence.

Table 9: Q4 variation across K/V training seeds (%). HumanEval averages three generation runs per fit; RULER64K uses one greedy run on 650 examples. Summary: mean ± sample SD across three fits.

Training seed	HumanEval	RULER64K
1234	94.92	82.99
1235 (main)	96.54	84.48
1236	96.54	85.21
Mean ± SD	96.00 ± 0.94	84.23 ± 1.13

The first three panels use the same BF16 reference input at each layer and measure K/V reconstruction error in the original coordinates, KL divergence from reference to quantized attention, and post-W_O output error. In the fourth panel, each method's hidden states independently pass through attention, residual connections, normalization, and MLPs in preceding layers before comparison with the reference output at that layer. Model projections and MLPs use BF16; rotations and attention use FP32 diagnostic reference computation. Reconstruction uses positions $[64 : -256]$; attention and output statistics start at position 320. Relative MSE is the sum of squared errors divided by reference energy.

Quantization configurations. Plain INT2, OSCAR, and both RRC-KV configurations use channel groups of 128, clipping ratio 0.96, BF16 metadata, and exact prefix-64/recent-256 cache windows. The OScaR official curve is a diagnostic reference constructed from its official algorithm: keys use Hadamard transforms and per-token L2 normalization with token groups of 32; values use channel groups of 32, integer zero points, nearest-even rounding, and BF16 metadata. The cache is packed in 256-token blocks, with original-coordinate reference values in the residual region. Values and output projections retain offline Hadamard folding and BF16 weight rounding. Key preprocessing calls the official CUDA extension, while dequantization and attention use reference computation. Packed blocks have been numerically checked against the native pack/decode interfaces.

Natural-text diagnostics outside calibration. We also use two predetermined excerpts from *Pride and Prejudice*, each comprising the first 8,192 tokens of an existing 16K out-of-calibration segment, covering all 36 layers and five methods. The first 7,168 tokens serve as BF16 prefill; the remaining 1,024 natural-text tokens are teacher-forced. Local diagnostics fix the teacher input at each layer, whereas propagation diagnostics carry hidden states through preceding layers; both replay the same natural-text sequence. Reconstruction, local-output, and propagated-state relative MSEs are computed by dividing pooled squared errors by pooled reference energy. KL is averaged over query/head observations. Table 10 reports the natural-text results, and Figure 2 reports the fixed AIME sequence.

Table 10: Q4 on two held-out 8K natural-text trajectories, all 36 layers. Lower is better. Reconstruction, local post-W_O output and propagated hidden-state errors are relative MSE; KL averages query/head observations.

Method	K	V	Attn. KL	Local output	Hidden state
INT2	0.701883	0.177470	1.357490	0.242603	0.174102
OSCAR	0.056050	0.154137	0.250272	0.081780	0.027227
OScaR official	**0.022930**	0.163967	0.170965	0.049974	0.024834
RRC-KV w/o center	0.034440	**0.142445**	0.136991	0.021066	0.014902
RRC-KV	0.031842	**0.142445**	**0.128706**	**0.020754**	**0.012444**

OScaR official has lower K reconstruction error than RRC-KV in 35/36 layers, but higher V reconstruction error in every layer. Its local-output error is lower in only 2/36 layers, and its propagated-state error is higher in every layer. This diagnostic retains native key Hadamard/L2 preprocessing, group size 32, and the 256-token eviction rule, with numerically checked diagnostic dequantization. Native checks use tolerances; they are not bitwise-equivalence tests of the fused service. These natural-text results support the distinction between K reconstruction and attention-error rankings, without isolating a specific error mechanism.

G.1 Offline Configuration Selection outside Calibration

Hard/smooth range combinations and query weights are compared using offline readout error on out-of-calibration texts; the default configuration is then fixed for downstream evaluation.

The query-weight and hard/smooth-objective comparisons use the same Q4 QKV dump. Rotation training still uses Alice and RFC 9110. Offline diagnostics use two consecutive, non-overlapping 16,384-token segments from the start of the main text of *Pride and Prejudice*[3], and two consecutive, non-overlapping segments of the same length starting at Section 2, *Protocol Overview*, of RFC 8446[4]. Together these give four fixed BF16 trajectories and 65,536 tokens. Both comparisons cover all 36 layers and use the same 64 query positions per trajectory, with identical quantization, centering, and pre-/post-W_O MSE measurements and no resampling. The new segments share no exact contiguous 64-token windows with the training calibration texts. These texts are excluded from rotation training and used for offline selection and diagnostics.

Table 11 reports calibration metrics and out-of-calibration readout error. Across $\lambda \in \{0, 1, 10\}$ in $\mathcal{L}_K = P_{\text{raw}} + \lambda c_Q^2$, post-$W_O$ MSE varies by about 3%; we fix $\lambda = 1$. The four hard/smooth K/V combinations differ by less than 1.1% in post-W_O MSE; we use hard K and smooth V. Each combination uses its fitted rotations and matching centers; soft K + hard V combines existing tensors without retraining.

Table 11: Q4 offline configuration selection. Calibration metrics use Alice/RFC 9110; out-of-calibration MSE uses the same four fixed trajectories for both panels. Hard/soft denotes the range objective, not the quantizer.

λ	Calib. P_{raw}	Calib. c_Q^2	Post-W_O MSE ↓
0	54.82365	1.240886	0.011167664
1	54.82815	0.035112	**0.010839558**
10	54.82681	0.003108	0.011041773
K objective	**V objective**	**Pre-W_O MSE ↓**	**Post-W_O MSE ↓**
Hard	Soft	0.005717	**0.010840**
Hard	Hard	0.005717	0.010848
Soft	Soft	0.005654	0.010956
Soft	Hard	**0.005641**	0.010932

H Local-Attention Measurement Protocol

The Q4 diagnostics use the original K/V rotations and key center underlying the 81.02%/84.48% paired RULER comparison. The candidate value center is computed from the same fit's original Alice/RFC calibration capture. We evaluate all 36 layers on four fixed out-of-calibration BF16 trajectories: two consecutive 16,384-token passages each from *Pride and Prejudice* and RFC 8446. These are the texts used for the offline comparisons in Table 11; their trajectories are separate from RULER generations, and centers are not refitted on them.

For every layer and passage, we select 256 query positions by rounding an equally spaced grid from position 256 to the final token, covering all 32 query heads. The cache uses prefix 64, recent 256, and eight-token flushes, so the recent region contains 256–263 tokens. Only low-bit history receives INT2 encoding error. Reconstruction MSE averages the codec error over all token/head/coordinate rows; attention measurements apply those errors only to history visible to each query.

All paired calculations use the same CPU backend: BF16 rotations, BF16 rounding after center subtraction, FP32 clipping/code selection, BF16 metadata and staged decoding (subtraction then multiplication), FP32 attention and full W_O, and FP64 error accumulation. Decoded centers are restored before reading; for values this is equivalent to adding the center weighted by the history attention mass. Pre-W_O errors restore and concatenate all 32 query heads (4,096 coordinates); post-

[3] https://www.gutenberg.org/ebooks/1342

[4] https://www.rfc-editor.org/rfc/rfc8446.txt

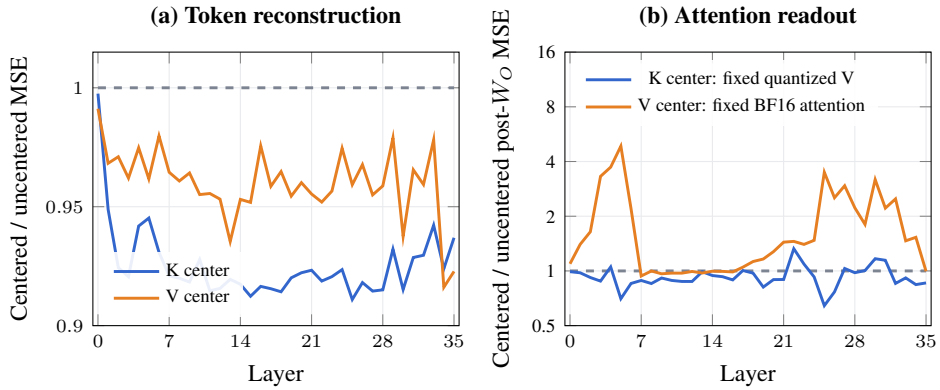


Figure 5: Centering diagnostics for the original Q4 fit on four out-of-calibration passages. Each layer averages passage-wise MSEs before taking centered/uncentered ratios. Left: K/V reconstruction. Right: K centering with fixed quantized V and V centering with fixed BF16 attention, both post-W_O. Dashed lines mark no change; the right axis is logarithmic.

W_O errors apply the full $4{,}096 \times 2{,}560$ projection. Both include finite-precision rotation/inverse-rotation error relative to the original BF16 trajectory.

K-only measurements keep BF16 values fixed. Joint measurements quantize both sides; the key-centering comparison fixes the same quantized, uncentered V. V-only measurements fix the original BF16 attention probabilities and change only value centering. The final joint comparison adds value centering to fixed key centering. For each condition, we average MSE or KL equally over the 36 layers and four passages, then take centered/uncentered ratios. These are fixed-trajectory readout diagnostics; paired task effects are measured separately in Appendix G.

H.1 Detailed Centering Measurements

Figure 5 and Table 12 separate reconstruction from attention readout. Key centering lowers reconstruction MSE by 7.35%, attention KL by 6.88%, and joint post-W_O MSE by 8.33%. Value centering lowers reconstruction MSE by 4.76% but raises fixed-attention pre-W_O and post-W_O MSE by 85.52% and 71.77%, respectively. The readout increase therefore precedes the output projection. Adding value centering to key centering increases joint post-W_O MSE by 19.64%.

Table 12: Centering effects on four out-of-calibration Q4 passages, original fit, all 36 layers. Ratios divide the mean metric with centering by its paired uncentered reference; lower is better. K-only uses BF16 V; V-only fixes BF16 attention; joint quantizes both.

Metric	Ratio	Change (%)
K reconstruction MSE	0.9265	−7.35
Attention KL, K center	0.9312	−6.88
K-only post-W_O MSE, K center	0.8987	−10.13
Joint post-W_O MSE, K center	0.9167	−8.33
V reconstruction MSE	0.9524	−4.76
V-only pre-W_O MSE, V center	1.8552	+85.52
V-only post-W_O MSE, V center	1.7177	+71.77
Joint post-W_O MSE, adding V center to K center	1.1964	+19.64

Low-range rows and selective reading. We predefine low-range rows using per-head lower-quartile thresholds of the original rotated V ranges, independently of reconstruction or output errors. Among history rows with nonzero squared attention weight, this group comprises 25.22% of rows but 59.77% of squared-weight mass. The threshold uses history positions read by the first KV head, followed by each head's nonzero-weight mask; ties can make the group exceed 25%. Squared

weights describe the diagonal term in Eq. (12), while complete readout also includes cross-row terms.

Table 13 switches selected rows back to uncentered INT2, retaining the same attention probabilities and BF16 windows. Protecting the low-range group removes the aggregate post-W_O increase; protecting its complement does not. Together with improved equal-row reconstruction, this intervention identifies a mismatch between the static calibration mean and the rows emphasized by attention.

Table 13: V-only selective-centering intervention under the same fixed-trajectory protocol. "Protected" rows use uncentered INT2, not BF16. All four conditions retain the original attention weights.

V encoding	Post-W_O MSE ($\times 10^{-3}$)
Uncentered	1.78481
Fully centered	3.06583
Centered, low-range group protected	1.76339
Centered, complement protected	3.08053

I Full Benchmark Results

I.1 Long-Context Evaluation

Table 14 gives the complete results across lengths for the final method and baselines on Q4 and Q8. Each entry is one greedy evaluation with a fixed fitted representation.

Table 14: RULER Full13 accuracy (%). Each model/method/length evaluates 13 tasks with 50 examples per task. Avg. equally weights four lengths. For Q8 at 64K, fitting and evaluation both use YaRN factor 4 and RoPE base 10^6. Other settings use native RoPE.

Model	Method	8K	16K	32K	64K	Avg.
Q4	BF16	97.86	97.39	95.51	92.97	95.93
	RRC-KV	95.80	93.52	88.15	84.48	90.49
	OSCAR	94.49	84.90	79.01	76.99	83.85
Q8	BF16	93.48	92.68	90.08	83.54	89.94
	RRC-KV	93.78	88.60	84.17	69.04	83.90
	OSCAR	92.19	85.63	78.47	61.21	79.38

J Related Baselines Reported in the Literature

To provide additional context on earlier representative methods, Tables 15 and 16 reproduce entries from Tables 1 and 2 of Zhou et al. (2026), retaining their full-precision references. The former contains previously reported KIVI AIME25 results collected in that paper (Liu et al., 2024b); the latter is its QuaRot-style Hadamard INT2 KV control (Ashkboos et al., 2024). These results retain the source configurations and scoring protocols, are presented separately from our measurements, and are excluded from the macro-average in Table 1. The source's five-task evaluation uses a 32K generation budget, temperature 0.6 with five runs for Qwen, and temperature 1.0 with three runs for GLM. Its LCB v6 results use the source evaluation set and protocol and are not labeled as our LCB Full1055.

Table 15: Literature-reported AIME25 accuracy at 32K generation, excerpted from Table 1 of Zhou et al. (2026). Values and BPE are retained as reported; the asterisk retains the source configuration label.

Method	BPE	Qwen3-8B	Qwen3-32B
BF16	16.00	66.00 ± 7.33	72.59 ± 7.41
KIVI-KV2	2.25	52.33 ± 9.00	57.41 ± 9.26
KIVI-KV2*	2.26	57.67 ± 9.00	59.05 ± 12.38
OSCAR	2.38	66.67 ± 3.33	74.00 ± 5.48

Table 16: Literature-reported five-task accuracy, excerpted from Table 2 of Zhou et al. (2026). Entries are mean $\pm$ SD over five runs for Qwen and three for GLM. QuaRot-INT2 denotes the source study's Hadamard-style KV baseline (2.25 BPE), not the full weight/activation quantization pipeline.

Model	Method	AIME25	GPQA	MATH500	HumanEval	LCB v6
Q4	BF16	74.67 ± 1.83	67.27 ± 1.80	93.55 ± 0.33	94.05 ± 0.54	48.66 ± 2.20
	QuaRot-INT2	0.00 ± 0.00	0.34 ± 0.48	5.67 ± 0.19	0.98 ± 0.10	0.00 ± 0.00
Q8	BF16	70.00 ± 3.33	56.67 ± 2.30	92.59 ± 0.62	85.95 ± 1.01	49.01 ± 2.13
	QuaRot-INT2	2.22 ± 1.57	14.98 ± 0.63	23.13 ± 1.88	9.80 ± 0.76	0.58 ± 0.58
Q32	BF16	68.67 ± 5.58	58.49 ± 1.21	93.55 ± 0.41	91.19 ± 1.11	59.06 ± 1.80
	QuaRot-INT2	0.00 ± 0.00	19.70 ± 1.09	17.60 ± 1.66	1.83 ± 0.33	0.39 ± 0.55
GLM	BF16	80.00 ± 3.33	73.23 ± 1.33	95.66 ± 0.61	91.46 ± 0.65	49.12 ± 0.59
	QuaRot-INT2	78.89 ± 1.92	68.01 ± 3.79	90.40 ± 0.35	89.87 ± 1.78	48.54 ± 4.79

K WHOLE-MODEL EFFICIENCY ACROSS HARDWARE PLATFORMS

Q4 is Qwen3-4B-Thinking-2507 with BF16 weights, running on one GPU with TP1. GLM-4.7 uses the same FP8 weights for all methods, with eight B200 GPUs forming one TP8 replica. The BF16 baseline refers to KV representation precision. All three methods use Triton attention, without FA4. OSCAR is a Triton-compatible adaptation of the fixed official algorithm, not RRC-KV with fusion disabled. Each method is measured over three independent service restarts. For each workload, two warm-up batches with 256 output tokens precede measurement with 1,024 output tokens per request; we report medians. The 16K/batch16 and 32K/batch8 settings are separate workloads, not a fixed-batch length sweep.

All methods use chunk size 8192, prefill-request1, a 400,000-logical-token pool, and graphs1/8/16/32, with radix, overlap, graph padding, and piecewise execution disabled. Low-bit methods use fixed assets, INT2 group size 128, clipping ratio 0.96, BF16 metadata, and prefix-64/recent-256 windows. For Q4, B200 uses Torch2.9.1+cu128/Triton3.5.1, while RTX uses Torch2.12+cu130/Triton3.7. All three methods share the software environment on each host; differences between hosts also include software effects.

Timing definitions. Let b be batch size, $n = 1024$ the output length per request, T batch latency, and T_{last} the time to the last request's first token. The official benchmark reports decode throughput as $bn/(T - T_{\text{last}})$ and end-to-end throughput as bn/T. Last TTFT includes queuing and communication. Per-request TTFT/TPOT comes from separate streaming diagnostics and is not included in official throughput timing. TPOT is computed as $(t_{\text{finish}} - t_{\text{first}})/(n - 1)$, averaged across requests, and then summarized by the median over three rounds. It is therefore not simply the reciprocal of the reported decode throughput.

Table 17: Q4 whole-model timing, medians of three independent service restarts. All rows use TP1 and 1,024 output tokens per request. Last TTFT and throughput use the native benchmark; TPOT uses separate streaming measurements.

GPU / input / batch	Method	Last TTFT (s)	Decode (tok/s)	E2E (tok/s)	TPOT (ms)
B200 / 16K / 16	BF16	7.9611	970.70	659.85	20.092
	OSCAR	8.4703	979.00	650.01	20.208
	RRC-KV	8.1672	976.69	656.46	20.103
B200 / 32K / 8	BF16	17.5045	498.28	241.31	23.541
	OSCAR	18.2650	483.54	232.67	24.352
	RRC-KV	17.9031	511.19	241.34	23.249
RTX / 16K / 16	BF16	23.4447	342.40	229.82	57.536
	OSCAR	23.8712	674.71	340.30	34.338
	RRC-KV	22.9407	666.20	344.97	34.182
RTX / 32K / 8	BF16	44.9776	186.51	92.15	62.165
	OSCAR	45.6343	359.62	119.74	41.359
	RRC-KV	41.9634	366.95	127.01	39.633

Table 18: GLM-4.7 FP8 weights on B200 TP8, three independent restarts, 1,024 output tokens/request. Decode and E2E are in tokens/s; request TTFT/TPOT are separate streaming diagnostics.

Input / batch	Method	Last TTFT (s)	Decode	E2E	Req. TTFT (s)	TPOT (ms)
16K / 16	BF16 KV	14.8462	601.61	389.36	7.9143	33.379
	OSCAR	16.0755	529.95	348.80	8.5715	37.632
	RRC-KV	15.6376	568.80	367.99	8.3773	35.211
32K / 8	BF16 KV	22.6364	306.75	166.05	12.7619	35.729
	OSCAR	23.5832	278.33	154.52	13.2347	38.658
	RRC-KV	23.0903	300.11	162.58	12.9785	36.458

Separate prefill measurements. Table 19 uses batch1/output1 and does not represent batch TTFT in the throughput workloads. Q4 uses four restarts per configuration, retaining three samples per round. GLM uses three restarts per configuration, with two warm-ups and three retained samples at each length. For each 8192-token chunk, CUDA events record forward time on all eight ranks; we take the rank maximum and sum across chunks. This includes gaps between forward submissions and collective-communication waits, but excludes scheduling and network TTFT. The two timing definitions are reported separately.

Table 19: Separate batch-1, one-output-token prefill diagnostics (ms). GLM uses the sum of per-chunk rank-maximum forward times on TP8; it is not the same timing primitive as Q4 single-GPU prefill.

Model / GPU	Input	BF16 KV	OSCAR	RRC-KV
Q4 / B200	16K	490.636	519.926	504.050
	32K	2177.074	2260.442	2223.726
Q4 / RTX PRO 5000	16K	1441.051	1478.823	1420.665
	32K	5568.997	5690.024	5215.167
GLM / B200 TP8	16K	918.205	998.476	974.397
	32K	2806.069	2924.868	2865.292

Implementation validation. Results for each hardware platform use separately frozen code and assets: Q4 on B200 uses a validated warp1 configuration, RTX uses the SM120 query-rotation route and fused value inverse, and GLM uses the TP8 fused path. Each implementation produces bitwise-identical token IDs, logprobs, and top-5 outputs in an 800-token on/off comparison. Performance percentages are computed from the corresponding experimental medians, with repeat counts given in each table. KV compression follows the representation-bit accounting in Table 8, whereas serving memory usage also includes preallocated cache pools.