RECAP: CALIBRATING LOW-BIT REASONING MODELS ON THE HISTORY THEY READ

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-bit quantization reduces the cost of serving reasoning models but can substantially degrade long chains of thought. Each step reads cached keys and values spanning thousands of tokens, allowing quantization errors to accumulate along the trajectory. Standard calibration operates on short segments that omit the earlier context needed by later reasoning steps. Even when this context is retained, cache reconstruction error alone does not capture how quantization errors affect subsequent attention. We present RECAP, a post-training quantization method that calibrates reasoning models on the history they read. Dynamic-prefix calibration separates long historical context from short-block reconstruction. It regenerates prefix keys and values with the current quantizer and rotates the active block to supervise every position along the trajectory. Prefix read geometry further supervises historical cache errors according to their effects on attention outputs, weighting key and value errors asymmetrically. With both components applied only during calibration, RECAP outperforms the baseline across multiple reasoning models and quantization settings. Under W4A4KV4, RECAP improves accuracy over the baseline by 13.3 percentage points on MATH500 with DeepSeek-R1-Distill-Qwen-1.5B and 15.8 points on AIME-120 with Qwen3-4B. These results highlight the importance of preserving historical dependencies during calibration for low-bit reasoning.

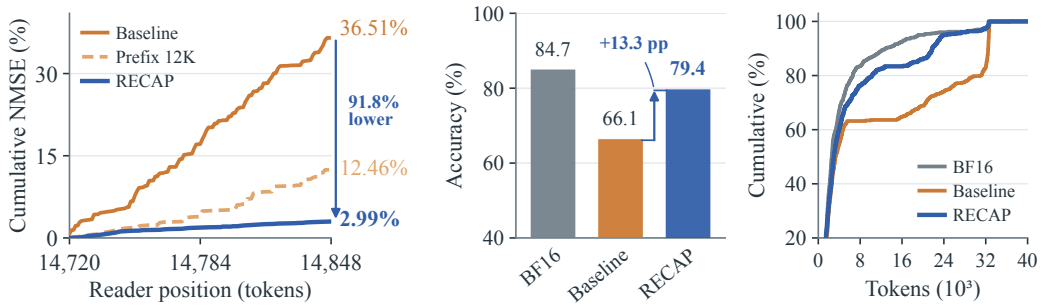
1 INTRODUCTION

Reasoning language models solve complex problems by generating extended sequences of intermediate steps (Wei et al., 2022; DeepSeek-AI, 2025; Qwen Team, 2025). These capabilities come with substantial inference costs, making low-bit quantization an attractive approach to deployment (Frantar et al., 2023; Lin et al., 2024; Xiao et al., 2023). However, low-bit quantization can substantially degrade reasoning performance (Liu et al., 2025a; Lv et al., 2026; Lee et al., 2026). Long-chain reasoning relies on repeated access to a growing history of intermediate states. Quantization errors in cached keys and values can distort attention outputs at later reasoning steps. As shown in Figure 1b,c, quantization can reduce answer accuracy while increasing generation length. This motivates calibration that preserves the historical dependencies underlying long-chain reasoning.

Existing post-training quantization methods seek to preserve model accuracy by improving the fidelity of quantized representations. QuaRot (Ashkboos et al., 2024) suppresses activation outliers through rotations, while FlatQuant (Sun et al., 2025) learns affine transformations through layer-output reconstruction. However, for reasoning models with low-bit weights, activations, and KV caches, calibration effectiveness also depends on how well it captures the role of earlier quantized states in later computation. Layer-output reconstruction captures these interactions within the supplied historical context and at the positions being reconstructed. Improving representation fidelity within this scope does not by itself ensure that longer historical dependencies are preserved across the trajectory. These considerations motivate examining which history calibration exposes, which reader positions it supervises, and how historical errors affect subsequent attention.

Our analysis identifies three challenges in preserving these historical dependencies. First, the cache-importance scores we tested transfer poorly across reasoning trajectories, limiting their usefulness for selecting which states to preserve. Calibration should therefore retain the historical context itself rather than a selected subset. Retaining this context raises a second challenge, which is to balance

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.



(a) Attention readout error **(b) Accuracy** **(c) Length**

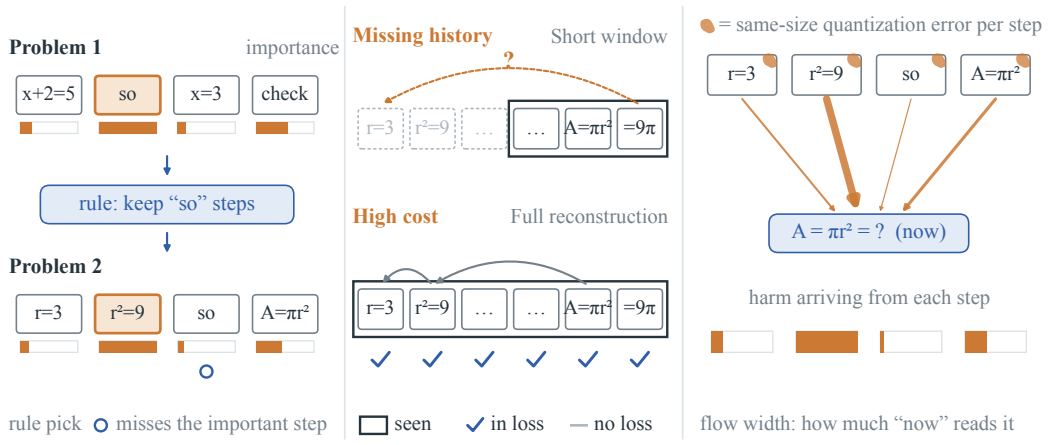
Figure 1: Attention error and reasoning performance of DeepSeek-R1-Distill-Qwen-1.5B. (a) Cumulative normalized attention error with a 12K-token INT4 KV prefix. (b,c) MATH500 accuracy and re-tokenized output-length distributions under W4A4KV4, with BF16 as the reference. Compared with the FlatQuant baseline, RECAP reduces final cumulative error by 91.8% and improves accuracy by 13.3 percentage points. It also produces fewer very long outputs, bringing the length distribution closer to BF16.

historical context, position coverage, and computational cost. Short windows omit relevant history, reconstructing only the final segment leaves earlier positions unsupervised, and reconstructing the full sequence at every update is costly. Third, even with adequate context and coverage, cache errors of similar magnitude can distort attention to different degrees, so raw cache reconstruction error remains an incomplete measure of their effect on later computation.

To address these challenges, we introduce RECAP (REad-geometry CAlibration with dynamic Prefixes), a post-training quantization framework that calibrates reasoning models on the history they read. *Dynamic-prefix calibration* addresses history and position coverage by allowing each short reconstruction block to read a long prefix regenerated with the current quantizer. Rotating the active block along the trajectory extends supervision to every position without reconstructing the entire sequence at each update. With this coverage in place, *prefix read geometry* further accounts for how errors in the retained history affect subsequent attention. It penalizes historical key and value errors according to their effects on attention outputs, complementing block reconstruction with direct supervision of the history being read.

Together, these components preserve historical dependencies during calibration without adding inference-time token selection or precision allocation. We evaluate RECAP across multiple reasoning models and quantization settings on mathematical reasoning, scientific question answering, and code generation. RECAP consistently matches or improves on the baseline. Under W4A4KV4, accuracy gains reach up to 13.3 percentage points on MATH500 and 15.8 points on AIME-120. Our contributions are as follows:

- **Motivation analysis.** We show that calibrating low-bit reasoning models is limited by historical dependencies: cache-importance scores transfer across trajectories at near-chance level, short-window and final-segment calibration each fail on the positions they leave uncovered, and cache errors of equal size affect later attention unequally.
- **Dynamic-prefix calibration.** We decouple the context used for calibration from the positions being reconstructed. At each update, the historical keys and values are regenerated with the current quantizer, and only a short block is reconstructed; this block is shifted along the trajectory across updates. Calibration therefore observes the long quantized history encountered at inference, and every position is supervised at the cost of short-block reconstruction.
- **Prefix read geometry.** We introduce an auxiliary cache objective derived from the local sensitivity of the attention output. Key errors are weighted by their projection onto query directions, and value errors by the attention they receive. Historical errors are thus penalized according to their influence on subsequent attention, which reconstruction error alone does not capture.



(a) Unpredictable importance **(b) Context and cost** **(c) Unequal influence**

Figure 2: Quantization sensitivity and calibration coverage. (a) A rule that finds the important cached step in one problem misses the important step in another. (b) Short calibration windows omit historical dependencies, while reconstructing every position in a long sequence incurs a higher computational cost per update. (c) The current step reads its cached history with unequal weight, so the same-size error on different steps has unequal influence.

2 Motivation

We first review reconstruction-based quantization and cached attention. We then examine three calibration challenges in reasoning models: limited transfer of importance scores, incomplete history and position coverage, and the mismatch between cache error and attention distortion.

2.1 Preliminaries

Low-bit quantization represents weights, activations, and cached keys and values with reduced precision. Reconstruction-based post-training quantization calibrates layer-specific parameters θ_ℓ on reference hidden states H^ℓ by minimizing

$$\min_{\theta_\ell} \mathbb{E}_{H^\ell}\left[\left\| f^\ell_{\theta_\ell}(H^\ell) - f^\ell_{\mathrm{FP}}(H^\ell)\right\|_F^2\right], \tag{1}$$

where f^ℓ_{FP} and $f^\ell_{\theta_\ell}$ denote the FP and quantized layers (Shao et al., 2024; Sun et al., 2025). The calibration sequences determine both the available context and the positions included in this objective.

Within one attention head, query q_s reads causally available keys k_t and values v_t through

$$a_{st} = \frac{\exp\left(q_s^\top k_t/\sqrt{d_h}\right)}{\sum_{t'\le s}\exp\left(q_s^\top k_{t'}/\sqrt{d_h}\right)}, \qquad y_s = \sum_{t\le s} a_{st}\, v_t, \tag{2}$$

where d_h is the head dimension. During autoregressive reasoning, earlier K/V remain in the cache and can influence later steps whenever they are read.

2.2 Quantization Sensitivity and Calibration Coverage

Limited transfer of importance scores. As illustrated in Figure 2(a), an importance score that works for one reasoning problem may not identify the important cached states in another. We restore individual quantized cache units to full precision and measure how much each repair reduces the teacher-forced loss of the final answer. Across four reasoning trajectories, scores fitted on three trajectories and evaluated on the held-out one rank the most beneficial 10% of repairs close to chance, with AUC between 0.43 and 0.50. The importance patterns observed in one reasoning

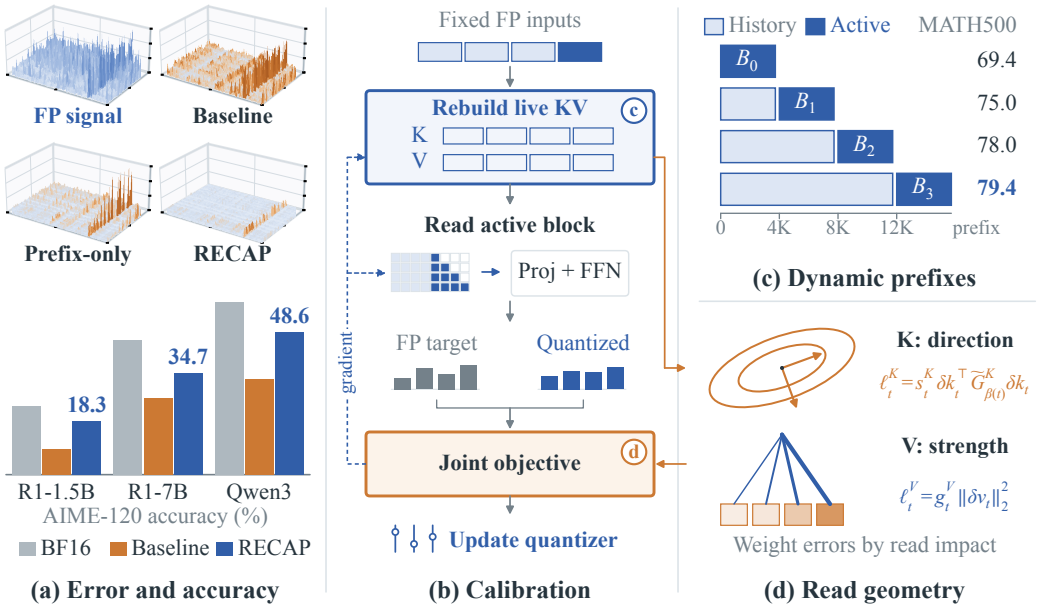


Figure 3: RECAP overview. (a) Full-precision attention magnitudes alongside quantization errors for the baseline, prefix-only calibration, and RECAP in one reader window; bars give W4A4KV4 AIME-120 accuracy from Table 1 on R1-1.5B, R1-7B, and Qwen3-4B. (b) The calibration workflow rebuilds historical keys and values with the current quantizer and combines block reconstruction with a read-geometry penalty. (c) Dynamic prefixes retain historical context as the active block rotates along the reasoning trajectory. (d) Read geometry weights key errors by direction and value errors by read strength, as in Equation 10; prefix and head indices are omitted.

trajectory, therefore, do not consistently carry over to another. This suggests that the tested scoring rules provide limited guidance for selecting important cached states across problems.

Historical context, position coverage, and cost. Figure 2(b) illustrates a trade-off in calibrating long reasoning sequences. Short windows can omit earlier premises that later steps depend on, preventing calibration from capturing how those steps use the preceding history. Extending the window restores access to more of this context, but reconstructing every position requires more computation per update as the sequence grows longer. Long reasoning trajectories, therefore, make full-sequence reconstruction costly. Supplying a long prefix while reconstructing only the final segment restores history, but leaves earlier query positions outside the objective. In teacher-forced diagnostics over an 8K context, a model calibrated on 4K windows diverges from the FP model late in the sequence, with KL rising from 0.14 nats/token in the first 1K positions to 7.33 in the last 1K. A model that reconstructs only positions 4K–8K with a 4K prefix fails in the opposite region, reaching 6.58 in the first 1K and 0.17 in the last. The challenge is to preserve the historical information needed by later steps and supervise every position without reconstructing the entire sequence in every update.

Unequal influence of cache errors. Figure 2(c) illustrates how quantization errors of the same size can have different effects on the current reasoning step. The current query reads the cached history with unequal attention weights. In the example, it relies more heavily on the step $r^2 = 9$ than on the connecting word "so", so errors at these positions can contribute differently to the attention output. Measuring cache error alone does not capture this difference (Zhou et al., 2026). Although layer-output reconstruction captures these effects at supervised positions, an additional cache loss should also reflect how later queries read the history.

These observations suggest three requirements for calibration. It should retain the historical context that later steps read, extend supervision across the trajectory without reconstructing the full sequence at every update, and account for how cache errors affect subsequent attention outputs.

3 RECAP

This section presents RECAP, with its overall design illustrated in Figure 3. We first establish the calibration setting by describing the reference reasoning trajectories and the layer reconstruction objective. We then introduce dynamic prefixes to preserve historical context during reconstruction, followed by prefix read geometry to account for how cache errors affect later attention outputs.

3.1 Calibration Formulation

We build a calibration set using reasoning trajectories generated by the original full-precision model (Williams et al., 2025):

$$\mathcal{D}_{\text{self}} = \{(x_i, r_i)\}_{i=1}^{N}. \tag{3}$$

Here, x_i is an input problem and r_i is the model's generated reasoning and answer. We tokenize them together, preserve their causal order and original positions, and cache the full-precision layer inputs H_i^ℓ and outputs $Y_i^\ell = f_{\text{FP}}^\ell(H_i^\ell)$. Calibration optimizes FlatQuant's learnable transformations, diagonal scales, and clipping parameters (Sun et al., 2025), while pretrained weights remain fixed.

For quantization parameters θ_ℓ, let $\widehat{Y}_i^\ell = f_{\theta_\ell}^\ell(H_i^\ell)$ denote the quantized layer output. Full-sequence reconstruction minimizes its mean squared error relative to the full-precision output:

$$\mathcal{L}_\ell^{\text{full}}(\theta_\ell) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{i,\ell}^{\text{full}}(\theta_\ell), \qquad \mathcal{L}_{i,\ell}^{\text{full}}(\theta_\ell) = \frac{1}{T_i d}\left\|\widehat{Y}_i^\ell - Y_i^\ell\right\|_F^2, \tag{4}$$

where T_i is the sequence length and d is the layer output dimension. Calibration is performed layerwise with teacher forcing, keeping the full-precision inputs and reference outputs fixed while optimizing θ_ℓ.

3.2 Dynamic Prefixes

Each calibration step reconstructs one block while retaining its preceding K/V. We partition a trajectory of length T into blocks $B_j = [jB, (j+1)B)$ of length B, and omit the trajectory index i below, so $\mathcal{L}_\ell^{\text{full}}$ denotes the per-trajectory loss $\mathcal{L}_{i,\ell}^{\text{full}}$. For an active block beginning at $P = jB$, we generate its prefix from the fixed FP layer inputs:

$$C_{<P}(\theta_\ell) = \text{KV}_{\theta_\ell}\big(H_{<P}^\ell\big). \tag{5}$$

The KV operator applies the current layer normalization, projection, activation, and cache quantization rules with the original positional transformations. We rebuild this prefix at every microstep to keep it consistent with the changing quantization parameters.

Queries in the active block attend to the prefix and causally available K/V within the block. We reconstruct only the active block's outputs:

$$\mathcal{L}_\ell^{\text{block}}(\theta_\ell; j) = \frac{1}{Bd}\sum_{t\in B_j}\left\|\widehat{Y}_t^\ell\big(H_{B_j}^\ell, C_{<P}(\theta_\ell)\big) - Y_t^\ell\right\|_2^2. \tag{6}$$

Gradients flow through both the active block and its prefix K/V. The block loss can therefore update the parameters that produce historical representations without computing the prefix's query, attention, or feed-forward outputs.

To supervise every position, we rotate the active block across the $J = T/B$ blocks: at epoch e, the sample with stable index i uses block $j = (e+i) \bmod J$, so each sample visits all J blocks once in every J epochs. The first block has an empty prefix and uses an ordinary causal forward pass.

At fixed parameters, the block losses partition the full-sequence objective. When the prefix is generated with the same transformations, quantization rules, and original positions as the full forward pass, each active-block output equals the corresponding full-sequence output, so

$$\frac{1}{J}\sum_{j=0}^{J-1}\mathcal{L}_\ell^{\text{block}}(\theta_\ell; j) = \mathcal{L}_\ell^{\text{full}}(\theta_\ell). \tag{7}$$

A uniformly sampled block, therefore, gives an unbiased estimate of Equation 4 and, because gradients flow through the prefix K/V, of its gradient. Each update computes queries, attention outputs, and feed-forward outputs only for the active block. The identity requires token-wise quantization statistics and the same causal support; parameter updates between blocks mean that a full rotation is not identical to full-sequence optimization.

3.3 Prefix Read Geometry

The block loss in Equation 6 measures errors at active outputs. Although this loss already reaches the prefix K/V through gradients, that signal is indirect: it is mixed with weight and activation errors at the active outputs and reflects only the queries of the current block. We therefore add a *prefix read geometry* loss that supervises each historical entry directly, weighting its error by how strongly later queries read it. Its metric is collected once from the original FP target model and frozen before layerwise calibration.

Attention sensitivity. Suppressing layer and head indices, let $q_s, k_t, v_t \in \mathbb{R}^{d_h}$ be FP reference attention tensors in physical coordinates: Q/K are post-RoPE, and reversible cache transformations have been removed. Let a_{st} and y_s denote the attention weights and head output in Equation 2. For fixed queries and perturbations only at prefix positions $t < P$, the first-order change is

$$\delta y_s = \sum_{t<P} a_{st}\delta v_t + \sum_{t<P} \frac{a_{st}}{\sqrt{d_h}}(v_t - y_s)q_s^\top \delta k_t. \tag{8}$$

Thus a key error acts through both its alignment with a query and the value contrast $v_t - y_s$. A value error enters directly, scaled by a_{st}.

For one-half the mean squared Euclidean distortion at the attention head output, the local generalized Gauss–Newton (GGN) is the attention Jacobian's Gram matrix. Retaining its individual K and V diagonal blocks gives

$$G^K_{t,P} = \mathbb{E}_{s\in\mathcal{Q}_P}\left[\frac{a_{st}^2\,\|v_t - y_s\|_2^2}{d_h}\,q_s q_s^\top\right], \qquad g^V_{t,P} = \mathbb{E}_{s\in\mathcal{Q}_P}\left[a_{st}^2\right], \tag{9}$$

where $\mathcal{Q}_P$ contains assistant query positions in the active block $[P, P+B)$, and the value block is $g^V_{t,P} I$. For grouped-query attention (GQA), we sum over all query heads sharing a KV head. The metric describes local head-output distortion before the output projection and omits cross-position and K–V cross terms.

Calibration objective. We retain per-entry key strength $s^K_{t,P} = \mathrm{tr}(G^K_{t,P})$ and value strength $g^V_{t,P}$. Unit-trace key matrices are pooled across fit trajectories and prefix conditions within each position bucket and KV head. Their shared approximation $\widetilde{G}^K_{\beta(t)}$, where $\beta(t)$ denotes the bucket containing position t, combines low-rank principal directions with an isotropic discarded-spectrum residual. This is a pooled approximation to Equation 9, not a full matrix per entry.

To compare quantized and FP caches, first undo the current learned transformations. In row-tensor notation, define $\Delta K = \widehat{K}'_\theta T^{-1}_{K,\theta} - K_{\mathrm{FP}}$ and $\Delta V = \widehat{V}'_\theta T^{-1}_{V,\theta} - V_{\mathrm{FP}}$. FP targets use the same fixed normalized layer inputs and original positions. The true inverses retain gradients. This avoids penalizing a change of cache basis as if it were quantization error.

Writing $\delta k_{t,h}, \delta v_{t,h}$ for the corresponding error vectors at KV head h, we define the read-geometry penalty and combine it with block reconstruction:

$$\begin{aligned}\mathcal{R}_\ell &= \frac{1}{H_{\mathrm{KV}}P}\sum_{h=1}^{H_{\mathrm{KV}}}\sum_{t<P}\left[s^K_{t,P,h}\,\delta k_{t,h}^\top \widetilde{G}^K_{\beta(t),h}\,\delta k_{t,h} + g^V_{t,P,h}\,\|\delta v_{t,h}\|_2^2\right], \\ \mathcal{L}_\ell &= \mathcal{L}^{\mathrm{block}}_\ell(\theta_\ell; j) + \alpha_\ell\,\mathcal{R}_\ell.\end{aligned} \tag{10}$$

Here, α_ℓ is a fixed layer-specific coefficient. For $P = 0$, $\mathcal{R}_\ell$ is defined to be zero. After calibration, neither the reference statistics nor the auxiliary loss is needed for inference.

4 EXPERIMENTS

We first compare a range of models and quantization methods across diverse benchmarks, then examine paired generation behavior. We assess both task accuracy and generation length to characterize the effects of quantization on reasoning. Ablation studies assess historical-prefix length and the contribution of read geometry beyond MSE.

4.1 SETUP

Models and benchmarks. We compare DeepSeek-R1-Distill-Qwen-1.5B/7B and Qwen3-4B, using BF16 as the unquantized reference and QuaRot and FlatQuant as quantization baselines. We evaluate MATH500 (Hendrycks et al., 2021; Lightman et al., 2023), AIME-120, GSM8K (Cobbe et al., 2021), GPQA-Diamond (Rein et al., 2023), and LiveCodeBench (Jain et al., 2024), covering mathematical reasoning, scientific question answering, and code generation. We report task scores throughout and mean generation length in the MATH500 and AIME-120 ablations.

Quantization and calibration. We evaluate W4A4KV4 and W4A8KV4 with symmetric per-channel weights, asymmetric per-token activations, asymmetric KV groups of 128, and 16-bit queries. Calibration uses target-model FP trajectories; the 1.5B ablation pool contains 1,000 NuminaMath-1.5 traces (Li et al., 2024). The default uses 64 fit trajectories, a 12K maximum prefix, a rotating 4K block (16K total; 1K = 1,024 tokens), and 32 epochs (512 updates/layer). AdamW uses base learning rate 5×10^{-3}, weight decay 0.01, and cosine decay. We fix α_ℓ at 0.1 times the lower median block-to-auxiliary gradient-norm ratio, computed at initialization over the first eight fit samples and all nonempty prefixes. The auxiliary multiplier is zero for the first 10% of updates, rises linearly to one over the next 10%, and then stays at one.

Evaluation protocol. Local generation uses temperature 0.6, top-p 0.95, and a 32,768-token output limit. AIME-120 combines 90 problems from 2022–2024 and 30 from 2025. Scores are percentages, averaged equally across the five benchmarks. In the ablations, MATH500 and AIME-120 scores are averaged over three evaluations per configuration.

4.2 MAIN RESULTS

Table 1 summarizes the three-model comparison under both quantization settings. Published R1 baselines follow Liu et al. (2025a, Table 1), and Qwen3-4B BF16 and W4A4KV4 FlatQuant follow Lv et al. (2026, Table 9); the remaining entries are our full-dataset evaluations, averaged over three runs. The released 7B QuaRot K4 checkpoints degenerate during generation; we report their measured low scores.

Consistent gains over FlatQuant. RECAP matches or exceeds FlatQuant on every model, bit setting, and benchmark in Table 1; the only tie is R1-7B on GSM8K under W4A4KV4 (90.9%). Under W4A4KV4, its average score is 6.3, 4.8, and 6.1 points higher than FlatQuant on R1-1.5B, R1-7B, and Qwen3-4B, respectively; under W4A8KV4, the corresponding gains are 6.8, 3.6, and 1.1 points. Relative to BF16, RECAP retains 89.0–91.2% of the average score under W4A4KV4, compared with 78.0–82.6% for FlatQuant, closing 41–58% of FlatQuant's remaining gap. Across all six model–precision combinations, RECAP achieves the highest average score among the quantized methods.

Gains concentrate on long multi-step reasoning. The largest improvements appear on AIME-120, MATH500, and LiveCodeBench, which require long reasoning chains or code generation. Under W4A4KV4, AIME-120 improves by 9.7, 8.6, and 15.8 points across the three models, and LiveCodeBench by 4.6, 6.4, and 8.3 points. On Qwen3-4B, the AIME-120 improvement recovers 60.5% of FlatQuant's deficit to BF16. In contrast, the gains on GSM8K and GPQA-Diamond never exceed 3.1 and 2.1 points in either setting. This pattern is consistent with our motivation that errors in historical K/V matter most when later steps repeatedly read a long history, although the table alone does not isolate this mechanism.

Table 1: Comparison of quantization methods on three reasoning models under W4A4KV4 and W4A8KV4. Scores are percentages, with BF16 (gray) as the reference. MATH, AIME, GSM, GPQA, and LCB denote MATH500, AIME-120, GSM8K, GPQA-Diamond, and LiveCodeBench; Avg. is their unweighted mean. Bold marks the best quantized method per model and setting. R1-1.5B/7B: DeepSeek-R1-Distill-Qwen-1.5B/7B.

		W4A4KV4						W4A8KV4					
Model	Method	MATH	AIME	GSM	GPQA	LCB	Avg.	MATH	AIME	GSM	GPQA	LCB	Avg.
R1-1.5B	BF16	84.7	23.3	84.5	36.2	16.4	49.0	84.7	23.3	84.5	36.2	16.4	49.0
	QuaRot	1.4	0.0	0.9	12.0	0.0	2.9	0.6	0.0	0.7	22.2	0.0	4.7
	FlatQuant	66.1	8.6	76.8	32.0	7.5	38.2	67.1	9.6	80.4	32.8	7.8	39.5
	RECAP	**79.4**	**18.3**	**79.9**	**32.8**	**12.1**	**44.5**	**82.1**	**22.1**	**80.9**	**34.3**	**11.9**	**46.3**
R1-7B	BF16	93.9	46.1	91.2	51.0	36.7	63.8	93.9	46.1	91.2	51.0	36.7	63.8
	QuaRot	1.4	0.0	0.4	24.6	0.0	5.3	1.0	0.0	0.6	22.4	0.0	4.8
	FlatQuant	84.1	26.1	**90.9**	46.6	12.4	52.0	88.4	35.8	90.0	48.0	23.1	57.1
	RECAP	**91.0**	**34.7**	**90.9**	**48.7**	**18.8**	**56.8**	**93.0**	**43.3**	**90.1**	**50.0**	**26.9**	**60.7**
Qwen3-4B	BF16	95.3	58.9	94.5	56.1	48.4	70.6	95.3	58.9	94.5	56.1	48.4	70.6
	QuaRot	89.8	29.2	91.8	45.5	23.1	55.9	95.0	56.7	**93.5**	43.4	42.5	66.2
	FlatQuant	89.9	32.8	92.1	47.5	29.1	58.3	94.4	57.5	92.8	50.0	45.5	68.0
	RECAP	**94.2**	**48.6**	**92.8**	**49.0**	**37.4**	**64.4**	**95.2**	**58.3**	**93.5**	**50.5**	**47.8**	**69.1**

Gains are largest where the baseline degrades most. FlatQuant degrades most on R1-1.5B, retaining 78.0% and 80.6% of the BF16 average under W4A4KV4 and W4A8KV4. This model also shows the largest average gains, 6.3 and 6.8 points, including 13.3 and 15.0 points on MATH500. When FlatQuant is already close to BF16, as for Qwen3-4B under W4A8KV4 (96.3%), the gain narrows to 1.1 points. QuaRot collapses on both R1 models, with averages of at most 5.3%, but remains competitive on Qwen3-4B, so the difficulty of low-bit quantization differs markedly across model families.

4.3 Ablation studies

Effect of prefix length. Table 2 compares 8K and 12K prefixes with a 4K active block on R1-1.5B under W4A4KV4. This increases total calibration coverage from 12K to 16K while retaining short-block reconstruction, so later query blocks can access a longer quantized history. For RECAP, the 12K configuration improves reported MATH500 and AIME-120 accuracy by 1.2 and 1.6 percentage points, respectively, while reducing mean generation length by 14.3% and 10.8%. Block MSE follows the same trend. These results favor the 12K configuration, although differences in calibration cohorts and update budgets limit attribution to prefix length alone.

Effect of read geometry. Block MSE already retains the prefix; read geometry adds direct supervision of historical K/V according to their effects on subsequent attention outputs. Key errors alter attention weights, whereas value errors perturb the content being aggregated. The reported read-geometry runs use statistics collected from the original FP model. Across both prefix lengths, RECAP improves reported accuracy by 1.0–1.6 points on MATH500 and 2.3–3.3 points on AIME-120, with 4.8–11.8% fewer generated tokens. The gains therefore combine higher accuracy with shorter generation. These point estimates do not establish statistical superiority, and decoding differences further limit causal attribution at 12K.

Generation length and correctness. Figure 4 compares BF16 and FlatQuant on the same 500 MATH500 questions. Lengths are re-tokenized from saved outputs, with the dashed line marking the nominal 32K budget. Of the 101 questions that change from correct to incorrect, 96 produce longer quantized outputs, often approaching the generation limit. This association shows why generation length must be read together with correctness: long outputs can still end in wrong answers. Accordingly, the shorter generations in Table 2 are evaluated alongside their accuracy gains, without attributing errors to length alone.

Calibration	Prefix	Acc. (%)	Tokens
MATH500			
Block MSE	8K	77.2	8,862
RECAP	8K	78.2	7,816
Block MSE	12K	77.8	7,120
RECAP	12K	79.4	6,697
AIME-120			
Block MSE	8K	14.4	24,234
RECAP	8K	16.7	22,279
Block MSE	12K	15.0	20,866
RECAP	12K	18.3	19,865

Table 2: Prefix and objective ablations on MATH500 and AIME-120 (R1-1.5B, W4A4KV4). Prefix: maximum history before the 4K block; Tokens: mean generation length.

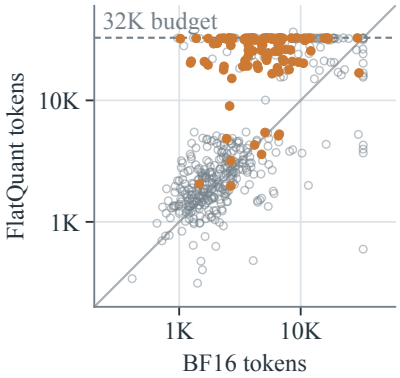


Figure 4: Paired MATH500 generation lengths for DeepSeek-R1-Distill-Qwen-1.5B: BF16 vs. FlatQuant W4A4KV4. Orange marks correct-to-incorrect cases; open gray points mark all others.

5 Related Work

Post-training quantization of LLMs. LLM PTQ combines sensitivity-aware quantization, outlier suppression, and reconstruction-based calibration. GPTQ (Frantar et al., 2023) uses approximate second-order information for weight quantization, while AWQ (Lin et al., 2024) protects salient channels through activation-aware scaling. SmoothQuant (Xiao et al., 2023) redistributes quantization difficulty from activations to weights, while QuaRot (Ashkboos et al., 2024) and SpinQuant (Liu et al., 2025b) suppress outliers through fixed and learned rotations, respectively. For reconstruction-based calibration, OmniQuant (Shao et al., 2024) learns equivalent transformations and weight clipping, and FlatQuant (Sun et al., 2025) learns affine transformations through layer-output reconstruction. Building on FlatQuant, RECAP extends calibration along the sequence dimension, preserving long historical context while distributing reconstruction across short blocks.

KV cache quantization. KV cache quantization balances memory savings, attention quality, and decoding overhead (Lin et al., 2025). KIVI (Liu et al., 2024) exploits key/value asymmetry with per-channel key and per-token value quantization, while KVQuant (Hooper et al., 2024) combines sensitivity-weighted quantization with outlier isolation. Outlier-aware rotations further improve low-bit cache representations (Su et al., 2025), and vector codebooks provide an alternative to scalar quantization (Li et al., 2025). Beyond cache distributions, OSCAR (Zhou et al., 2026) uses attention-aware covariance statistics to construct rotations and clipping thresholds.

Quantization and calibration for reasoning models. For reasoning models, quantization sensitivity varies across model families and tasks (Liu et al., 2025a), motivating reasoning-aware training (Okoshi et al., 2026; Lee et al., 2026). In post-training compression, calibration quality depends on both data selection (Williams & Aletras, 2024) and sequence length (Oh & Oh, 2025). Self-calibration generates calibration text with the model itself, avoiding dependence on external corpora (Williams et al., 2025). AYOT (Wang et al., 2026) uses reasoning traces and final answers from the high-precision target model alongside calibration questions.

6 Conclusion

We presented RECAP, a post-training quantization method that calibrates low-bit reasoning models on the history they read. Dynamic-prefix calibration brings the long quantized history into short-block reconstruction, and prefix read geometry supervises historical cache errors by their effect on attention, both only during calibration. Across multiple reasoning models, benchmarks, and quantization settings, RECAP consistently improves on the baseline, with the largest gains on long-horizon reasoning, although the remaining gap to full precision varies across models.

ETHICS STATEMENT

This study does not involve human participants. Potential biases, inaccurate outputs, and misuse should be considered when applying language models. Performance on standard benchmarks does not guarantee reliability across all use cases. Deployment in sensitive settings requires additional assessment and appropriate human oversight.

REPRODUCIBILITY STATEMENT

We retain experiment configurations, evaluation outputs, and analysis scripts to support verification of the reported results. These records document the settings and procedures used for the experiments and analyses. Reproduction should account for differences in software environments and the variability introduced by stochastic evaluation.

AI USE STATEMENT

AI tools were used to assist with manuscript revision and language refinement, including improvements to wording, clarity, and organization. We also used the full-precision target models to generate reasoning trajectories for self-calibration. The authors are responsible for the scientific content, the interpretation of the results, and the accuracy and integrity of the final manuscript.

REFERENCES

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b5b939436789f76f08b9d0da5e81af7c-Abstract-Conference.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://github.com/deepseek-ai/DeepSeek-R1.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.17323.

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, 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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2401.18079.

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, 2024. URL https://arxiv.org/abs/2403.07974.

Janghwan Lee, Sihwa Lee, Jinseok Kim, Yongjik Kim, Jieun Lim, Jinwook Oh, and Jungwook Choi. ReQAT: Achieving full-precision reasoning accuracy with 4-bit floating-point quantization-aware training. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.15682v1.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. Hugging Face dataset repository, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-1.5. NuminaMath-1.5 dataset card; accessed 11 September 2026.

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.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.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. In *Proceedings of Machine Learning and Systems*, volume 7, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/hash/fbe2b2f74a2ece8070d8fb073717bda6-Abstract-Conference.html.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? an empirical study on quantized reasoning models. In *Conference on Language Modeling*, 2025a. URL https://arxiv.org/abs/2504.04823v2. Version 2.

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. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e5b1c0d4866f72393c522c8a00eed4eb-Abstract-Conference.html.

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.

Keyu Lv, Manyi Zhang, Xiaobo Xia, Jingchen Ni, Shannan Yan, Xianzhi Yu, Lu Hou, Chun Yuan, and Haoli Bai. What makes low-bit quantization-aware training work for reasoning LLMs? a systematic study, 2026. URL https://arxiv.org/abs/2601.14888v1. Version 1; Table 9.

Jaehoon Oh and Dokwan Oh. Beyond fixed-length calibration for post-training compression of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 19355–19366. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1054. URL https://aclanthology.org/2025.findings-emnlp.1054/.

Yasuyuki Okoshi, Hikari Otsuka, Daichi Fujiki, and Masato Motomura. Towards quantization-aware training for ultra-low-bit reasoning LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/209423f076b6479ab3a4f45886e30306-Abstract-Conference.html.

Qwen Team. Qwen3-4B model card, 2025. URL https://huggingface.co/Qwen/Qwen3-4B. Original Qwen3-4B checkpoint; accessed 8 September 2026.

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, 2023. URL https://arxiv.org/abs/2311.12022.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

Zunhai Su, Hanyu Wei, Zhe Chen, Wang Shen, Linge Li, Huangqi Yu, and Kehong Yuan. RotateKV: Accurate and robust 2-bit KV cache quantization for LLMs via outlier-aware adaptive rotations. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 6200–6208. International Joint Conferences on Artificial Intelligence Organization, 2025. doi: 10.24963/ijcai.2025/690. URL https://www.ijcai.org/proceedings/2025/690.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness matters for LLM quantization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57587–57613. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25l.html.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, and Anbang Yao. Attend to your own thoughts: Breaking the barrier for post-training quantization of reasoning LLMs through the lens of 1.58-bit quantization, 2026. URL https://arxiv.org/abs/2608.01078v1.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2201.11903.

Miles Williams and Nikolaos Aletras. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10100–10118. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.544. URL https://aclanthology.org/2024.acl-long.544/.

Miles Williams, George Chrysostomou, and Nikolaos Aletras. Self-calibration for language model quantization and pruning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10149–10167. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.509. URL https://aclanthology.org/2025.naacl-long.509/.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

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, 2026. URL https://arxiv.org/abs/2605.17757v1.

A Implementation and Calibration Details

Calibration data and budgets. The R1-1.5B pool contains 1,000 FP trajectories from NuminaMath-1.5, split into 900 source-train and 100 source-validation records. Filtering removes malformed responses and normalized exact-match overlaps with MATH500 and GSM8K; trajectories are used regardless of answer correctness. Fit cohorts retain leading segments of length-eligible trajectories. Table 3 gives the two relevant calibration budgets. Their cohorts differ, so the prefix comparison also changes data exposure.

Table 3: Calibration budgets. Both configurations use a 4K active block, optimizer batch 4, and microbatch 1. Each position is reconstructed five times at 8K prefix and eight times at 12K.

Configuration	Max prefix	Total context	Fit samples	Epochs	Updates/layer
Historical ablation	8K	12K	128	15	480
Default	12K	16K	64	32	512

Transformation and diagonal parameters use the base learning rate in Section 4.1; learned clipping uses a tenfold multiplier. Gradients are accumulated with detached loss normalization. Both gradient norms used to set α_ℓ cover the same trainable parameters, with unused gradients treated as zero, and are measured without updating the initialization. The default recipe collects geometry from the original FP model. Calibration preserves causal order and original position IDs. At every microstep, prefix K/V are regenerated from fixed FP layer inputs using the current quantization parameters, with gradients retained through prefix construction.

Query sampling. For each sample and nonempty prefix, divide eligible assistant queries in the active block into four position strata and sample up to 16 queries without replacement per stratum. A sampled query from stratum a receives weight $N_a/(n_a|\mathcal{Q}_P|)$, where N_a and n_a are its eligible and sampled counts. These inverse-inclusion weights estimate the mean in Equation 9. Query selections remain fixed across layers and GQA heads.

Shared key directions. Let $\widehat{G}^K_{i,t,P,h}$ be a sampled key block and $s^K_{i,t,P,h}$ its trace. For each 4K position bucket β and KV head h, pool positive-trace blocks across fit samples and prefix conditions:

$$\overline{G}^K_{\beta,h} = \frac{1}{|\mathcal{I}_{\beta,h}|} \sum_{(i,t,P)\in\mathcal{I}_{\beta,h}} \frac{\widehat{G}^K_{i,t,P,h}}{s^K_{i,t,P,h}}. \tag{11}$$

Entry strengths remain sample- and prefix-specific. If $\lambda_1 \geq \cdots \geq \lambda_{d_h}$ are the pooled eigenvalues and U_r contains the leading eigenvectors, use $r = 8$ and

$$\rho = \frac{1}{d_h - r} \sum_{j=r+1}^{d_h} \lambda_j, \qquad \widetilde{G}^K = \rho I + U_r \operatorname{diag}(\lambda_1 - \rho, \ldots, \lambda_r - \rho) U_r^\top. \tag{12}$$

This PSD approximation preserves the pooled trace and replaces discarded directions with an isotropic residual. Values retain scalar read strengths. Cache errors use the current true right inverses with gradients, as in Section 3.3.

Prefix computation and cost. Active-block outputs match the corresponding full-forward outputs when the layer inputs, cache transformations, original positions, tokenwise quantization statistics, and causal support are identical and prefix gradients are retained. Each update computes $BP + B(B+1)/2$ query–key pairs, compared with $T(T+1)/2$ for full-sequence reconstruction; with $T = 4B$, this is about a quarter of the full-sequence pairs on average over the four blocks, and feed-forward outputs are computed only for the B active positions. Prefix K/V are still projected at every update, the rank-r penalty adds $O(PH_{\mathrm{KV}}d_h r)$ work, and read-geometry statistics are collected once per model before calibration.

B EVIDENCE FOR THE MOTIVATION

Transfer of importance scores. We label the top 10% of cache repairs by their reduction in teacher-forced answer loss. Each of four folds fits scores on three trajectories (1,536 candidates) and evaluates on the held-out trajectory (512 candidates); clusters are fitted on training trajectories only. Table 4 reports all five tested scores. None exceeds an AUC of 0.51, including the future-read count, which uses information unavailable during calibration. These are four discovery trajectories from an earlier V-codeword configuration, so the results support the limited-transfer observation for these scores, not a general impossibility claim or a matched W4A4KV4 comparison.

Table 4: Importance-score transfer, averaged over four held-out trajectories. AP is average precision; P@10% is precision among the selected 10%. Future-read count uses information unavailable during calibration and is computed from a separate trace; it serves as a reference.

Score	AUC	AP	P@10%
Codeword shrinkage	0.495	0.109	0.096
Residual-direction cluster	0.504	0.108	0.111
Position-bin shrinkage	0.429	0.091	0.053
Future-read count	0.495	0.107	0.091
Local V error	0.504	0.103	0.067

Position coverage. The 8K diagnostic evaluates eight development trajectories in their original causal context, averaging FP-to-quantized KL within 1K position buckets. Short-window and fixed-tail calibration fail in opposite regions, whereas full 8K calibration stays between 0.140 and 0.190 nats/token in every bucket. Rotating-prefix calibration over a 12K context, evaluated on four development trajectories, likewise stays between 0.101 and 0.191 nats/token, so no region of the sequence is left unsupervised. Different cohorts prevent direct magnitude comparisons between these diagnostics; they do not measure task accuracy or isolate rotation from all other calibration changes.