TRACEKV: TRAJECTORY-GUIDED SIGNED ERROR FEEDBACK FOR KV-CACHE QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-bit key-value (KV) cache quantization reduces the storage requirements of large language models, but quantization errors can alter their output distributions and generation trajectories. Because errors across tokens can cancel or reinforce one another, reducing individual reconstruction errors alone does not directly control their combined effect on generation. We propose TRACEKV, a method that uses self-generated traces to guide cache reconstruction through signed error feedback while preserving the base quantizer's encoding format, bit widths, and metadata footprint. TRACEKV generates a short continuation with the bfloat16 (BF16) model, then replays it to compute gradients of multi-step Kullback-Leibler (KL) divergence with respect to cached values. Within each layer and attention head, it accumulates signed projections of quantization errors onto these gradients and selects between independent quantization and residual-feedback reconstruction for each token, encouraging positive and negative contributions to offset. Trace replay then selects among the completed candidate caches. Across four quantization representations, TRACEKV reduces cumulative trajectory distortion. On PatternKV and OSCAR, gains persist for 64 tokens beyond the H64 selection window; paired ablations support retaining error signs. In two evaluations with frozen data and protocols, generation restarts from the compressed prompt. With KVarN as the base quantizer, TRACEKV increases the mean length of the generated prefix that exactly matches the BF16 continuation by 14.4 and 22.8 tokens over independent KVarN, respectively, with prefix length capped at 1024 tokens.

1 INTRODUCTION

Low-bit KV caches reduce the memory required by autoregressive transformers, but quantization errors can change the tokens they generate. Existing methods improve the quantized representation through asymmetric grouping (Liu et al., 2024), rotations (Su et al., 2025; Zhou et al., 2026), pattern residuals (Zhang et al., 2025), and variance normalization (Muller et al., 2026), or allocate precision across cache components (Li et al., 2025; Liu et al., 2026; Zhang et al., 2026). We study a complementary question: *with the representation, bit allocation, and retained tokens fixed, can choosing different reconstructions better preserve generation?*

Output-aware quantization already recognizes that an error matters through the computation that reads it (Zhou et al., 2026; Yun et al., 2026; Fernández-Menduiña et al., 2026; Laus et al., 2026). Our focus is how errors combine across tokens. A local loss gradient projects each token's reconstruction error onto a signed scalar. Opposing projections can cancel in their sum, an interaction that magnitude-only scores discard. Error feedback offers a way to influence this sum by adding a preceding token's quantization residual before quantizing the next token. The challenge is deciding when to apply feedback using a signal relevant to the model's subsequent generation.

TRACEKV obtains this signal from a short greedy continuation generated by the same BF16 model. Lookahead generation has been used to inform cache eviction (Wang et al., 2025b); here, replaying the same token sequence measures output-distribution distortion and supplies its gradients with respect to cached values. The controller maintains a signed ledger, a running sum of gradient–error projections within each layer and attention head. For each token, it chooses independent quantization or residual feedback to keep the ledger close to zero. Completed caches are then evaluated by replay, which selects among guarded reconstructions, independent quantization, and temporal feedback.

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.

These choices preserve the base quantizer's code alphabet, bit-width, group shape, and metadata footprint. Codes, scales, and offsets may change; persistent storage does not. SQuat also exploits reconstruction freedom, using a prompt query subspace to compensate key errors (Wang et al., 2025a). Our controller instead uses signed, multi-step value gradients. Construction requires neither training nor task labels. The trace, gradients, and controller state are temporary; only the selected cache remains for decoding.

Experiments test both reconstruction fidelity and generation. Frozen stress tests cover affine, KVarN, PatternKV, and OSCAR's rotation/clipping component. Paired ablations on two sources favor the signed ledger over its magnitude-only counterpart. When generation restarts from the compressed prompt, mean prefix retention capped at 1024 tokens improves over independent KVarN by 14.4 and 22.8 tokens on two frozen sources with 4096-token generation budgets. These gains arise from reconstruction choices within the same quantized representation and persistent storage budget.

Our contributions are a reconstruction controller combining self-generated traces with signed error feedback at fixed storage; paired ablations that test the role of signed contributions; and an evaluation relating short-horizon distortion to free-running prefix retention. Together, these results show that input-specific reconstruction choices can improve trajectory fidelity without allocating additional persistent memory.

2 Finite-Horizon Fidelity and Signed Error

2.1 Measuring generation fidelity

A frozen model maps a prompt x to a clean KV cache C. Given a cache reconstruction $\widehat{C}$ with fixed code widths, group shapes, and metadata count, we measure its effect on the model's output distributions. A greedy rollout from C provides tokens $y_{1:H}$ and reference logits z_t. We replay the same tokens from $\widehat{C}$, allowing each cache to evolve through its own computations, and sum the per-step KL divergence:

$$D_H = \sum_{t=1}^{H} \mathrm{KL}(p(z_t) \,\|\, p(\hat{z}_t)), \qquad p(z) = \mathrm{softmax}(z). \tag{1}$$

This finite-horizon objective measures output-distribution fidelity without task labels or model-weight updates (Dutta et al., 2024; Jiang & Jin, 2026).

2.2 Using signed sensitivity to shape cache error

The effect of a value error depends on how the model responds to it. Let $C_0 = (\widehat{K}_0, \widehat{V}_0)$ be the independently quantized cache, with $\widehat{V}_0 = V + E_0$, and hold its initial keys fixed. Around this operating point, a value change δ changes the objective to first order by

$$L(\widehat{V}_0 + \delta) - L(\widehat{V}_0) = \sum_i \langle g_i, \delta_i \rangle + O(\|\delta\|^2), \qquad g_i = \nabla_{V_i} L\big|_{\widehat{V}_0}. \tag{2}$$

The projection $\langle g_i, \delta_i \rangle$ gives each token's signed contribution to the local change in output loss. Contributions from different tokens can reinforce or cancel.

TraceKV uses these projections to maintain a running signed ledger while constructing the cache. Its controller chooses the magnitude of the accumulated projection, rather than performing greedy descent on the signed Taylor change. The ledger is therefore a construction rule; replay of each completed cache measures the finite-horizon objective used for final selection.

2.3 Why the trace horizon matters

The sensitivity direction depends on the evaluated horizon, so gradients from a short trace may not rank errors the same way at a longer horizon. We therefore use the same horizon for cache construction and candidate selection, then evaluate generation beyond that horizon (§4.5).

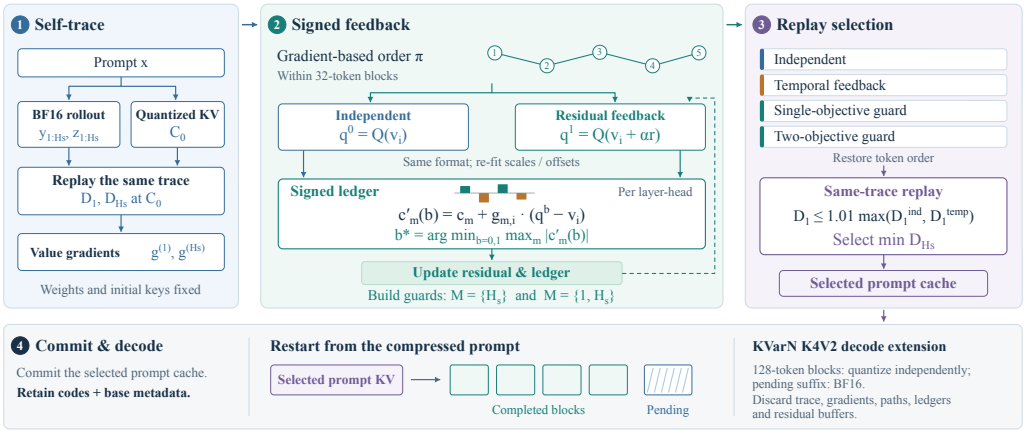

Figure 1: Four stages of TRACEKV: a BF16 self-trace supplies value gradients; signed feedback constructs candidates in the base format; same-trace replay selects the prompt cache; and the selected codes and metadata are committed before decoding. In the evaluated KVarN extension, complete decode blocks are quantized independently while the pending suffix remains BF16.

3 TRACEKV: SIGNED ERROR SHAPING AT FIXED STORAGE

Given a BF16 prefill cache and a base quantizer, TRACEKV returns a quantized cache with the same code widths, layout, and metadata count as that base. It builds the cache in three stages: obtain value sensitivities from a self-generated trace, use signed error feedback to construct candidates, and replay them to select the final cache. It then commits the selected prompt cache for decoding (Figure 1).

3.1 TRACE-DERIVED VALUE SENSITIVITY

After prefill, the BF16 model generates a greedy trace of H_s tokens ($H_s = 32$ by default and 64 in the horizon-aligned experiments) and records its logits. We independently quantize the cache to obtain C_0, then replay the trace from C_0 and differentiate the one-step and cumulative KL objectives, D_1 and D_{H_s}, with respect to the cached values. Model weights and the initial quantized keys are fixed. The resulting gradients indicate which value-error directions most affect the trace's output distributions.

3.2 CONSTRUCTING VALUES WITH SIGNED ERROR FEEDBACK

For one layer–head stream, let v_i be the clean value at token i, $\mathcal{Q}_i$ the base quantize–dequantize operation, and r_{t-1} the residual passed from the preceding token in processing order π. At step t, the controller forms two legal reconstructions:

$$q^{(0)}_{\pi_t} = \mathcal{Q}_{\pi_t}(v_{\pi_t}), \qquad q^{(1)}_{\pi_t} = \mathcal{Q}_{\pi_t}(v_{\pi_t} + \alpha r_{t-1}), \qquad \alpha = 0.5. \tag{3}$$

The first rounds the value independently; the second first adds part of the incoming residual. Both use the base quantizer's code alphabet, group shape, and metadata count. The fitted scale and offset values may differ because each candidate input is quantized separately.

For each objective $m \in \{1, H_s\}$, the controller normalizes the gradient within each layer, batch item, and KV head over all trace positions and value channels:

$$\bar{g}^{(m)}_{\ell bhi} = \frac{g^{(m)}_{\ell bhi}}{\max(\|g^{(m)}_{\ell bh}\|_F, 10^{-20})}. \tag{4}$$

For each candidate, the signed projection updates its objective-specific ledgers. The controller chooses the candidate with the smallest maximum absolute ledger:

$$c_t^{(m,b)} = c_{t-1}^{(m)} + \left\langle \bar{g}_{\pi_t}^{(m)}, q_{\pi_t}^{(b)} - v_{\pi_t} \right\rangle, \tag{5}$$

$$b_t = \arg \min_{b \in \{0,1\}} \max_{m \in \{1, H_s\}} |c_t^{(m,b)}|. \tag{6}$$

The chosen reconstruction is committed to this candidate cache, and its remaining rounding error becomes the residual for the next token:

$$r_t = v_{\pi_t} + \alpha b_t r_{t-1} - q_{\pi_t}^{(b_t)}. \tag{7}$$

Thus the ledger tracks a scalar signed effect on the output objective, while the residual is a value-shaped vector used to form the next feedback candidate. If independent quantization is chosen, the incoming residual is discarded and the current rounding error is carried forward.

The multi-horizon guard uses both D_1 and D_{H_s}. The single-objective guard uses only D_{H_s}. Each layer and KV head maintains its own residual and ledger.

3.3 Processing order for residual feedback

Since each residual affects the next token's candidate, the token order can change the reconstructed values. Within each 32-token block, TraceKV starts from the block's first token, then repeatedly appends the unvisited token whose D_{H_s} gradient is nearest in Euclidean distance to the current token's gradient, after concatenating gradients across layers for each batch item and KV head. Layers share this deterministic order, while maintaining separate residuals and ledgers. After construction, values return to their original token positions; the path is not part of the stored cache. Temporal-order comparisons test this path in §4.4.

3.4 Replay-based selection of a completed cache

The local ledger guides token-level choices; replay evaluates each completed cache under the model's full finite-horizon objective. The candidate set contains independent quantization, temporal residual feedback, the single-objective guard, and the multi-horizon guard. We replay each candidate on the BF16 self-trace and measure D_1 and D_{H_s}.

Let B_1 be the larger D_1 of the independent and temporal baselines. Candidates with $D_1 \leq 1.01 B_1$ are eligible, and among them we select the candidate with the smallest measured D_{H_s}, breaking ties by D_1 and then a fixed order. Both baselines are always eligible. Thus, on the selection trace, the chosen cache's D_1 is at most 1.01 times the larger baseline value, and its D_{H_s} is no greater than either baseline's.

3.5 Applying the controller to base quantizers

For a transformed base representation, write its encode–decode operation as $\mathcal{R}_{\phi,i}$, with frozen structural parameters ϕ. The controller applies feedback in the original value coordinates:

$$\hat{v}_i^{(b)} = \mathcal{R}_{\phi,i}(v_i + b\alpha r_{t-1}), \qquad i = \pi_t. \tag{8}$$

This lets TraceKV change value reconstructions while retaining the base quantizer's stored format. In KVarN, the Hadamard rotation and tilewise normalization are fixed while row scales and offsets are refitted. PatternKV retains its centers, assignments, and residual mask while re-quantizing centered residuals. The OSCAR experiment applies the controller to its rotation-and-clipping quantize–dequantize component. Across these compositions, code counts, scalar counts, widths, and layouts match the corresponding base representation (Appendix F).

3.6 Cache output and decoding

The selected prefill cache stores only the base-format codes and metadata; the trace, gradients, path, residuals, ledgers, and unselected candidates are discarded. Keys use the base quantizer, while the controller shapes values. In the evaluated decode extension, newly completed 128-token blocks are

Matched base	Len.	Build/select	D_{64}/base	D_{128}/base	Obs. max	Func.
Affine K4V2	1024	H32/H32	0.748	–	1.000@64	6/8
Affine K4V2	2048	H32/H32	0.738	–	0.853@64	8/8
Affine K4V2	4096	H32/H32	0.581	–	0.842@64	7/8
Affine K4V2	8192	H32/H32	0.666	–	1.032@64	7/8
Affine (14B)	1024	H32/H32	0.950	–	1.068@64	4/4
KVarN K4V2	1024	H32/H32	0.614	–	0.887@64	7/8
KVarN (14B)	1024	H32/H32	0.832	–	1.242@64	4/4
PatternKV K4V2	1024	H64/H64	0.515	0.729	0.855@128	8/8
OSCAR codec INT2	1024	H64/H64	0.938	0.962	1.000@128	3/4

Table 1: Frozen results on sources isolated from preceding development within each base, relative to each row's independent matched base (geometric means, lower is better; 8 cells per row, 4 for the single-model 14B and OSCAR rows). For H64/H64 rows, D_{64} is the construction-and-selection score and D_{128} is a cumulative evaluation that includes this window; D_{128} is never used for selection. "–" marks an unevaluated horizon; "$x@h$" is the observed maximum per-cell ratio at horizon h, not a bound. "Func." counts cells selecting a functional guard. The OSCAR row uses its rotation/clipping codec with sink $= 0$, recent $= 0$.

quantized independently and the incomplete suffix remains BF16 under the base's pending-block rule.

4 EXPERIMENTS

4.1 SETUP

We ask whether TRACEKV reduces trajectory distortion across quantizers, whether this improves free-running generation, and which design choices explain the gains. Our main models are Qwen3-4B (Yang et al., 2025) and DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI et al., 2025), with BF16 references. Table 1 adds separately frozen DeepSeek-R1-Distill-Qwen-14B tests and an OSCAR-codec test on Qwen3-4B-Thinking-2507 with public calibration. Prompts place self-contained math, code, instruction-following, or continuation tasks after task-irrelevant WikiText (Merity et al., 2017).

Hyperparameters are chosen on development data. Before each confirmation, we freeze and hash the model, code, data, seed, and selection rule, excluding task identities and background rows used in preceding development within that base. Mechanism ablations reuse KVarN confirmation inputs and are repeated on a fresh source (§4.4). A *cell* pairs one prompt with one model; a *source* is a fixed, hash-pinned set of input identities. K4V2 denotes 4-bit keys and 2-bit values.

Within each representation, we compare independent quantize–dequantize (QDQ), temporal first-order error feedback, and the two functional guards at matched code and persistent-metadata budgets: affine K4V2 with group size 64; the official KVarN and PatternKV recipes at 3.375 and 3.793 effective bits; and the OSCAR rotation/clipping codec at G128 INT2, 2.50 bits. The OSCAR comparison uses public calibration with no protected prefix or recent tokens (sink $= 0$, recent $= 0$), isolating the codec component of its serving configuration. Candidate selection uses teacher-forced D_H (Eq. 1); outcome evaluation measures free-running prefix retention and paired task correctness. Distortion ratios below one favor TRACEKV.

Analysis. Comparisons are paired by prompt and model. Unless stated, we report two-sided exact sign tests on discordant pairs and paired-bootstrap intervals.

4.2 TRAJECTORY DISTORTION ACROSS FOUR REPRESENTATIONS

Affine K4V2. At 1024 tokens, TRACEKV reduces D_{64} by 25.2% against independent quantization and 35.8% against temporal feedback. Against temporal feedback, the frozen length-extension ratios are 0.642/0.734/0.564/0.519 at 1024/2048/4096/8192 tokens. Gains over both baselines are larger at 4096–8192, though nonmonotone.

KVarN. TRACEKV reaches 0.614× independent KVarN and 0.588× temporal feedback at matched storage. At 14B, the ratio is 0.832× independent for KVarN and 0.950× for affine on the same prompts.

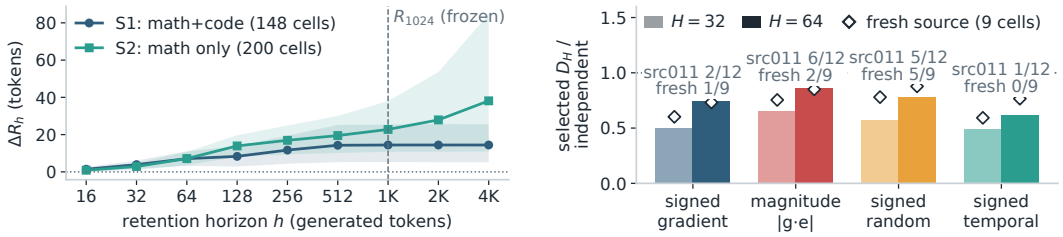

Figure 2: **Left:** prefix retention on the two frozen 4096-token sources (ΔR_h, TRACEKV+KVarN minus KVarN, tokens; paired-bootstrap 95% bands). R_{1024} is S1's pre-declared primary and S2's pre-declared continuity secondary (S2's frozen primary is accuracy, §4.3). **Right:** paired mechanism ablations (§4.4): bars are the 12 source-011 KVarN confirmation cells reused as a paired diagnostic set (src011), hollow markers the fresh 9-cell source—selected-cache D_H relative to independent KVarN at the H32 construction horizon and cumulative H64. H64 is not used for selection but includes the H32 construction window; labels count safety fallbacks to a baseline on each source.

4.3 FREE-RUNNING GENERATION: RETENTION AND ACCURACY

Let R_h count the initial generated tokens that match BF16 before the first mismatch, capped at h (Deiseroth et al., 2024); identical early termination is right-censored at h.

Prespecified outcomes. S1 (74 fresh MATH-500 (Hendrycks et al., 2021) and HumanEval (Chen et al., 2021) identities $\times$ two models $= 148$ cells) was frozen with R_{1024} retention as *primary* and accuracy as secondary. S2 (100 fresh math-only identities, 200 cells) was frozen after S1 to confirm its accuracy result, making paired task correctness its *primary*; retention is a prespecified secondary outcome with separate gates.

Retention improves on both sources. At 4096-token generation budgets (Figure 2, left), S1's primary R_{1024} improves by $+14.4$ tokens ($p = 1.1 \times 10^{-4}$), and S2's secondary R_{1024} improves by $+22.8$ ($p = 3.7 \times 10^{-4}$). Every frozen trajectory gate passes on both sources.

Task accuracy. On S1, accuracy improves by 9.5 points (95% interval $[+0.7, +17.6]$, $p = 0.049$); S2 shows a 2.5-point increase that does not meet its prespecified accuracy gate ($p = 0.53$). Prefix retention improves on both sources.

4.4 MECHANISM ABLATIONS AND COMPOSITION

Retaining signed contributions improves the controller. We replace each signed projection in the ledger with its absolute value and compare this magnitude-only control on the same 12 cells, sharing the trace, gradients, path, candidate-generation rule, and hardware. Dropping the sign loses roughly half of the improvement over independent KVarN at H64 (0.864 vs. 0.744; guard-level ratio $1.43\times$ at the construction horizon) and triples safety fallbacks (6/12 vs. 2/12). On the fresh 9-cell source, the ratio replicates (0.851 vs. 0.732; 1.162 vs. 1.161). The paired comparison isolates the effect of retaining token signs in the ledger. PM-KVQ uses elementwise magnitude sensitivity for bit allocation (Liu et al., 2026); our controller instead keeps the bit allocation fixed and balances signed token-level projections.

Composition with SQuat. We apply the value controller to a port of SQuat's subspace-orthogonal key quantizer (Wang et al., 2025a) on the same nine fresh prompt–model pairs. The value-side gain is similar under SQuat and KVarN keys: 0.713 vs. 0.732 at 4 bits, and 0.831 vs. 0.866 at 2 bits. Across six key-width and subspace-dimension settings, the gains differ by at most 0.055, supporting composition in these tests.

4.5 GENERALIZATION ACROSS REPRESENTATIONS AND HORIZONS

Held-out horizon extension. PatternKV and OSCAR are constructed and selected at H64, where their D_{64} ratios are 0.515 and 0.938. After 64 additional tokens, the cumulative D_{128} ratios are 0.729 and 0.962, respectively; both remain below one.

5 Related Work

Quantized representations and functional objectives. KV quantizers use grouping, normalization, and structured representations (Liu et al., 2024; Hooper et al., 2024; Zhang et al., 2025; Muller et al., 2026). OSCAR calibrates rotations from attention-aware covariance (Zhou et al., 2026); OptR optimizes rotations against post-projection attention-output error, including cross-token value-error interactions (Yun et al., 2026). Recent work learns non-uniform code levels and thresholds against the language-model objective using surrogate gradients (Anonymous, 2026). TraceKV keeps the base alphabet, layout, and bit allocation fixed, using per-input trace gradients to select cache reconstructions and metadata values at the same storage budget. Our OSCAR comparison covers only its rotation/clipping QDQ component (§3.5).

Using input-specific queries and continuations. SQuat controls key error against a prompt-derived query subspace without training, compensating unquantized coordinates as it rounds each key (Wang et al., 2025a). Our value controller instead uses multi-step output-loss gradients from a self-generated trace; its gains persist with our SQuat key port in the tested settings (§4.4). KVSculpt optimizes a shorter continuous cache by matching attention outputs and log-sum-exp values on retained and synthetic future queries; its evaluation already uses teacher-forced multi-step KL (Jiang & Jin, 2026). Future-aware eviction also uses generated or drafted continuations to select retained tokens (Wang et al., 2025b; Galim et al., 2026). TraceKV retains the base's token count and layout and uses its continuation to select value reconstructions at fixed storage. DGAP uses trained modules to restore local top-K logits during decoding; TraceKV instead uses self-trace gradients to choose cache values during construction (Dzikanyanga et al., 2026).

Sensitivity, allocation, and error feedback. PM-KVQ allocates bit-widths across Transformer blocks using elementwise absolute gradient–error sensitivity (Liu et al., 2026). Our controller keeps bit allocation fixed and accumulates signed token-level projections, using the sign pattern to balance contributions across tokens. Error compensation and coupled quantization objectives precede this work in GPTQ and QuIP (Frantar et al., 2023; Chee et al., 2023), alongside sigma-delta noise shaping (Benedetto et al., 2006; Wang, 2008). GPFQ uses data-dependent greedy path following for weight quantization (Lybrand & Saab, 2021). TraceKV brings signed feedback to per-input KV reconstruction: self-trace gradients guide the choices, and end-to-end replay evaluates the completed caches.

6 Limitations

Our evidence covers greedy decoding over finite horizons and a limited set of model and quantizer configurations; broader generation settings remain to be studied.

7 Conclusion

TraceKV uses self-generated traces to guide signed error feedback within a fixed KV quantization format and storage budget. Its controller balances gradient–error projections across tokens, then selects a cache by replaying the trace. Across four quantization representations, reconstruction control lowers short-horizon cumulative distortion; paired ablations support retaining error signs. On two frozen sources, generation from the compressed prompt preserves longer prefixes of the BF16 continuation. Together, these results show that choosing how errors combine is a useful degree of freedom for low-bit KV-cache quantization.

Reproducibility statement

The code supplement provides frozen protocols, model revisions, prompt manifests, immutable results, and scripts for running experiments and generating tables and figures. Confirmation inputs are separated from preceding development within each base quantizer. The appendices document data isolation, effective storage, execution environments, per-cell results, and the mechanism ablations.

AI USE STATEMENT

The core method was designed by the author. Generative AI tools assisted with manuscript drafting, translation, editing, and reference checking, and with experimental validation, including designing diagnostic comparisons, implementing and debugging experimental and analysis code, and interpreting results. The author retains responsibility for the methods, experiments, claims, attribution, and final content of this work.

REFERENCES

Anonymous. Beyond: Better-than-full-precision KV caches via learnable non-uniform quantization as implicit regularization, 2026. URL https://openreview.net/pdf?id=rhsvM1j5n3. Anonymous ACL submission.

John J. Benedetto, Alexander M. Powell, and Özgür Yılmaz. Sigma-delta quantization and finite frames. *IEEE Transactions on Information Theory*, 52(5):1990–2005, 2006. doi: 10.1109/TIT.2006.872849.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems*, 2023.

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.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948v1.

Björn Deiseroth, Max Meuer, Nikolas Gritsch, Constantin Eichenberg, Patrick Schramowski, Matthias Aßenmacher, and Kristian Kersting. Divergent token metrics: Measuring degradation to prune away LLM components – and optimize quantization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6764–6783, 2024. doi: 10.18653/v1/2024.naacl-long.377. URL https://aclanthology.org/2024.naacl-long.377/.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is not all you need. In *Advances in Neural Information Processing Systems*, 2024.

Gradwell Dzikanyanga, Yanqi Pan, Weihao Yang, Donglei Wu, Wen Xia, and Hao Huang. High-accuracy low-bit KV-cache quantization via local distribution restoration. *arXiv preprint arXiv:2607.16248*, 2026.

Samuel Fernández-Menduiña, Amir Ziashahabi, Eduardo Pavez, Antonio Ortega, and Salman Avestimehr. Spend bits where queries look: KV cache vector quantization with attention-preserving transforms. *arXiv preprint arXiv:2608.04074*, 2026.

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.

Kevin Galim, Ethan Ewer, Wonjun Kang, Minjae Lee, Hyung Koo, and Kangwook Lee. Draft-based approximate inference for LLMs. In *International Conference on Learning Representations*, 2026.

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. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

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*, 2024.

Bo Jiang and Sian Jin. KVSculpt: KV cache compression as distillation. *arXiv preprint arXiv:2603.27819*, 2026.

Hannah Laus, Claudio Mayrink Verdun, Hao Wang, Flavio du Pin Calmon, and Felix Krahmer. KV cache compression through the lens of transform coding. *arXiv preprint arXiv:2608.14191*, 2026.

Xing Li, Zeyu Xing, Yiming Li, Linping Qu, Hui-Ling Zhen, Yiwu Yao, Wulong Liu, Sinno Jialin Pan, and Mingxuan Yuan. KVTuner: Sensitivity-aware layer-wise mixed-precision KV cache quantization for efficient and nearly lossless LLM inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36451–36485. PMLR, 2025. URL https://proceedings.mlr.press/v267/li25dd.html.

Tengxuan Liu, Shiyao Li, Jiayi Yang, Tianchen Zhao, Feng Zhou, Xiaohui Song, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. PM-KVQ: Progressive mixed-precision KV cache quantization for long-CoT LLMs. In *International Conference on Learning Representations*, 2026.

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 *International Conference on Machine Learning*, 2024.

Eric Lybrand and Rayan Saab. A greedy algorithm for quantizing neural networks. *Journal of Machine Learning Research*, 22(156):1–38, 2021.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Lorenz K. Muller, Philippe Bich, Chiara Boretti, Hyun-Min Chang, Jiawei Zhuang, and Lukas Cavigelli. KVarN: Variance-normalized KV-cache quantization mitigates error accumulation in reasoning tasks. *arXiv preprint arXiv:2606.03458*, 2026.

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, 2025. doi: 10.24963/ijcai.2025/690. URL https://www.ijcai.org/proceedings/2025/690.

Hao Wang, Ligong Han, Kai Xu, and Akash Srivastava. SQuat: Subspace-orthogonal KV cache quantization. In *Conference on Language Modeling*, 2025a.

Yang Wang. Sigma-delta quantization errors and the traveling salesman problem. *Advances in Computational Mathematics*, 28(2):101–118, 2008. doi: 10.1007/s10444-006-9016-1.

Yixuan Wang, Shiyu Ji, Yijun Liu, Yuzhuang Xu, Yang Xu, Qingfu Zhu, and Wanxiang Che. Lookahead Q-cache: Achieving more consistent KV cache eviction via pseudo query. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34158–34174, 2025b. doi: 10.18653/v1/2025.emnlp-main.1732. URL https://aclanthology.org/2025.emnlp-main.1732/.

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.

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.

Ji Zhang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Xinglin Wang, Jiayi Shi, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. PatternKV: Flattening KV representation expands quantization headroom. *arXiv preprint arXiv:2510.05176*, 2025.

Tao Zhang, Ziqian Zeng, Hao Peng, Huiping Zhuang, and Cen Chen. MixKVQ: Query-aware mixed-precision KV cache quantization for long-context reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7189–7204. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.326. URL https://aclanthology.org/2026.acl-long.326/.

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.

A GATED WEIGHTED ABEL IDENTITY

This identity relates the signed quantization errors to the residuals used by the feedback update. It describes the controller's error bookkeeping; candidate quality is evaluated separately by trace replay.

Within one layer–head stream, let π be the processing order and let $b_t \in \{0, 1\}$ indicate whether feedback is selected. Set $\beta_t = \alpha b_t$, where $\alpha \in (0, 1]$. For a reconstruction q_{π_t} of $v_{\pi_t} + \beta_t r_{t-1}$, the residual update is $r_t = v_{\pi_t} + \beta_t r_{t-1} - q_{\pi_t}$. Thus $e_{\pi_t} = q_{\pi_t} - v_{\pi_t} = \beta_t r_{t-1} - r_t$.

Lemma 1 (Gated weighted Abel identity)**.** *For $r_0 = 0$, in exact arithmetic,*

$$\begin{aligned}\sum_{t=1}^{T} \langle g_{\pi_t}, e_{\pi_t} \rangle = & -\langle g_{\pi_T}, r_T \rangle \\ & + \sum_{t=1}^{T-1} \langle \beta_{t+1} g_{\pi_{t+1}} - g_{\pi_t}, r_t \rangle.\end{aligned} \tag{9}$$

This is a gated application of the summation-by-parts step used in sigma-delta analysis (Benedetto et al., 2006; Wang, 2008).

Proof. Substitute $e_{\pi_t} = \beta_t r_{t-1} - r_t$ and use $r_0 = 0$:

$$\sum_{t=1}^{T} \langle g_{\pi_t}, e_{\pi_t} \rangle = \sum_{t=1}^{T} \langle \beta_t g_{\pi_t}, r_{t-1} \rangle - \sum_{t=1}^{T} \langle g_{\pi_t}, r_t \rangle \tag{10}$$

$$= \sum_{t=1}^{T-1} \langle \beta_{t+1} g_{\pi_{t+1}} - g_{\pi_t}, r_t \rangle - \langle g_{\pi_T}, r_T \rangle. \tag{11}$$

When $b_t = 0$, the current independent-quantization error initializes the next residual; when $b_t = 1$, the incoming residual contributes with gain α. Thus the identity accounts for the actual gated update.

Implementation scope. The controller keeps separate ledgers for each objective, layer, batch item, and KV head, while using a shared processing path. Codes, fitted scalar values, and base-representation metadata form the cache output; the path, gradients, and ledgers guide reconstruction and are not part of the encoded cache.

B PROTOCOLS AND DATA ISOLATION

Table 2 lists the frozen protocols for the main results and paired mechanism diagnostics. "Cells" are prompt–model pairs; "Fam." is the number of task families. Table 3 gives the recorded identity and background-row isolation counts: each confirmation source shares zero task identities and zero WikiText background rows with every manifest that preceded it.

Experiment	Models	Cells	Fam.	Frozen	Status	Runner
Affine K4V2 1024 (row 1)	Q4B, L8B	8	4	2026-06-30	passed	`envelope.py`
Affine K4V2 2048 (row 2)	Q4B, L8B	8	4	2026-06-30	passed	`envelope_balanced.py`
Affine K4V2 4096 (row 3)	Q4B, L8B	8	4	2026-07-19	passed	`envelope_balanced.py`
Affine K4V2 8192 (row 4)	Q4B, L8B	8	4	2026-07-19	passed	`envelope_balanced.py`
Affine K4V2 14B (row 5)	Q14B	4	4	2026-07-19	passed	`envelope_balanced.py`
KVarN K4V2 1024 (row 6)	Q4B, L8B	8	4	2026-06-30	passed	`kvarn.py`
KVarN K4V2 14B (row 7)	Q14B	4	4	2026-07-19	passed	`kvarn.py`
PatternKV K4V2 H64 (row 8)	Q4B, L8B	8	4	2026-06-30	passed	`patternkv.py`
OSCAR rotation/clip codec INT2 H64 (row 9)	Q4B-T	4	4	2026-06-30	passed	`oscar.py`
S1 source-126 (retention primary)	Q4B, L8B	148	2	2026-07-19	passed	`kvarn_task_eval.py`‡
S2 source-128 (accuracy primary)	Q4B, L8B	200	math	2026-08-25	accuracy gate unmet	`kvarn_task_eval.py`‡
Path-default confirmation	Q4B, L8B, Q14B	9	3	2026-09-02	paired comparison	`kvarn_path_ablation.py`
Sign ablation (signed vs. magnitude)	Q4B, L8B, Q14B	12	4	2026-08-25	diagnostic	`kvarn_signflip_ablation.py`
Path-geometry ablation (3 arms)	Q4B, L8B, Q14B	12	4	2026-09-02	diagnostic	`kvarn_path_ablation.py`

Table 2: Frozen protocols and diagnostics behind the paper. Models: Q4B = Qwen3-4B, L8B = DeepSeek-R1-Distill-Llama-8B, Q14B = DeepSeek-R1-Distill-Qwen-14B, Q4B-T = Qwen3-4B-Thinking-2507. Dates are the recorded freeze (or run) dates; sign and path ablations are paired diagnostics on source-011 prompts. Runner files live in `experiments/kv_ideas/` with the prefix `run_gref_` dropped and are identified by matching the recorded SHA-256 against the repository; ‡the S1/S2 rows were produced by the revision of the named file pinned before the 2026-09-09 follow-up branch moved it (Appendix G, code revisions). The OSCAR rotation/clip codec row uses sink=0 and recent=0.

Source	Task overlap	Prior tasks	New tasks	Row overlap	Prior rows	New rows	Background
Affine K4V2 1024	0	40	4	0	1730	426	–
Affine K4V2 2048	0	40	4	0	1730	426	–
Affine K4V2 4096	0	358	4	0	2772	474	–
Affine K4V2 8192	0	358	4	0	2772	474	–
Affine K4V2 14B	0	40	4	0	1730	426	–
KVarN K4V2 1024	0	44	4	0	2156	370	–
KVarN K4V2 14B	0	44	4	0	2156	370	–
PatternKV K4V2 H64	0	72	–	0	402	446	WikiText-2:test
OSCAR rotation/clip codec INT2 H64	0	176	4	0	848	377	WikiText-2:test
S1 source-126	0	284	74	0	2848	1043	WikiText-2:train
S2 source-128	0	362	100	0	3891	707	WikiText-2:train
Path-default confirmation	0	–	–	0	–	–	WikiText-2:validation

Table 3: Recorded data isolation per frozen source. "Prior" counts the task identities and WikiText background rows used by all earlier manifests at freeze time; "New" counts those introduced by the source; overlaps are exact set intersections computed at freeze. The 4096- and 8192-token identities of source-127 share task suffix and background text, the longer prompt retaining more of the same WikiText passage. The 4096/8192 stress rows and the 14B columns reuse the hash-pinned source-127 and source-010/011 manifests, respectively. The background namespace is shown where the protocol records it (the June-30 affine and KVarN protocols record only the counts). The pinned S2 manifest updates prior tasks/prior rows/new rows from 284/2848/1043 to 362/3891/707; prompt identities and zero-overlap results are unchanged.

C PER-CELL RESULTS BEHIND TABLE 1

Tables 4 and 5 list every cell of Table 1: the selected candidate (Ind. = independent QDQ, Temp. = temporal error feedback, FG = single-objective functional guard, MHG = multi-horizon functional guard), whether the safety envelope fell back to a baseline, and the selected cache's D_{64} relative to the row's independent base. For the H64/H64 rows (PatternKV, OSCAR rotation/clip codec) D_{64} is the selection score and D_{128} is a cumulative endpoint that includes the construction window. Geometric means reproduce Table 1.

Model	Prompt	Selected	Fallback	D_{64}/Ind.	D_{128}/Ind.
Qwen3-4B	`math-010-l1024`	MHG	no	0.537	–
Qwen3-4B	`code-010-l1024`	MHG	no	0.898	–
Qwen3-4B	`instruction-010-l1024`	MHG	no	0.651	–
Qwen3-4B	`continuation-010-l1024`	MHG	no	0.541	–
R1-Llama-8B	`math-010-l1024`	FG	no	0.850	–
R1-Llama-8B	`code-010-l1024`	Ind.	yes	1.000	–
R1-Llama-8B	`instruction-010-l1024`	FG	no	0.681	–
R1-Llama-8B	`continuation-010-l1024`	Ind.	yes	1.000	–
Affine K4V2, 1024 (H32/H32): geometric mean			6/8 func.	**0.748**	–
Qwen3-4B	`math-010-l2048`	FG	no	0.748	–
Qwen3-4B	`code-010-l2048`	MHG	no	0.757	–
Qwen3-4B	`instruction-010-l2048`	FG	no	0.667	–
Qwen3-4B	`continuation-010-l2048`	FG	no	0.689	–
R1-Llama-8B	`math-010-l2048`	FG	no	0.821	–
R1-Llama-8B	`code-010-l2048`	MHG	no	0.683	–
R1-Llama-8B	`instruction-010-l2048`	FG	no	0.706	–
R1-Llama-8B	`continuation-010-l2048`	MHG	no	0.853	–
Affine K4V2, 2048 (H32/H32): geometric mean			8/8 func.	**0.738**	–
Qwen3-4B	`math-163-l4096`	FG	no	0.790	–
Qwen3-4B	`code-163-l4096`	FG	no	0.545	–
Qwen3-4B	`instruction-163-l4096`	Temp.	yes	0.230	–
Qwen3-4B	`continuation-163-l4096`	MHG	no	0.664	–
R1-Llama-8B	`math-163-l4096`	MHG	no	0.543	–
R1-Llama-8B	`code-163-l4096`	FG	no	0.539	–
R1-Llama-8B	`instruction-163-l4096`	FG	no	0.795	–
R1-Llama-8B	`continuation-163-l4096`	FG	no	0.842	–
Affine K4V2, 4096 (H32/H32): geometric mean			7/8 func.	**0.581**	–
Qwen3-4B	`math-163-l8192`	MHG	no	1.032	–
Qwen3-4B	`code-163-l8192`	MHG	no	0.828	–
Qwen3-4B	`instruction-163-l8192`	MHG	no	0.606	–
Qwen3-4B	`continuation-163-l8192`	MHG	no	0.601	–
R1-Llama-8B	`math-163-l8192`	FG	no	0.264	–
R1-Llama-8B	`code-163-l8192`	MHG	no	0.696	–
R1-Llama-8B	`instruction-163-l8192`	Ind.	yes	1.000	–
R1-Llama-8B	`continuation-163-l8192`	MHG	no	0.681	–
Affine K4V2, 8192 (H32/H32): geometric mean			7/8 func.	**0.666**	–
R1-Qwen-14B	`math-010-l1024`	MHG	no	0.945	–
R1-Qwen-14B	`code-010-l1024`	MHG	no	0.931	–
R1-Qwen-14B	`instruction-010-l1024`	MHG	no	0.867	–
R1-Qwen-14B	`continuation-010-l1024`	MHG	no	1.068	–
Affine K4V2, 14B (H32/H32): geometric mean			4/4 func.	**0.950**	–

Table 4: Affine K4V2 rows of Table 1 (source-010 at 1024/2048 and for the 14B column; source-127 at 4096/8192).

Model	Prompt	Selected	Fallback	D_{64}/Ind.	D_{128}/Ind.
Qwen3-4B	`math-011-l1024`	FG	no	0.859	–
Qwen3-4B	`code-011-l1024`	FG	no	0.515	–
Qwen3-4B	`instruction-011-l1024`	FG	no	0.598	–
Qwen3-4B	`continuation-011-l1024`	Temp.	yes	0.340	–
R1-Llama-8B	`math-011-l1024`	FG	no	0.493	–
R1-Llama-8B	`code-011-l1024`	FG	no	0.754	–
R1-Llama-8B	`instruction-011-l1024`	MHG	no	0.887	–
R1-Llama-8B	`continuation-011-l1024`	FG	no	0.679	–
KVarN K4V2, 1024 (H32/H32): geometric mean			7/8 func.	**0.614**	–
R1-Qwen-14B	`math-011-l1024`	FG	no	0.921	–
R1-Qwen-14B	`code-011-l1024`	FG	no	0.526	–
R1-Qwen-14B	`instruction-011-l1024`	MHG	no	1.242	–
R1-Qwen-14B	`continuation-011-l1024`	FG	no	0.795	–
KVarN K4V2, 14B (H32/H32): geometric mean			4/4 func.	**0.832**	–
Qwen3-4B	`math-023-l1024`	FG	no	0.635	0.695
Qwen3-4B	`code-023-l1024`	FG	no	0.540	0.715
Qwen3-4B	`instruction-023-l1024`	FG	no	0.318	0.725
Qwen3-4B	`continuation-023-l1024`	MHG	no	0.463	0.643
R1-Llama-8B	`math-023-l1024`	FG	no	0.461	0.781
R1-Llama-8B	`code-023-l1024`	FG	no	0.541	0.686
R1-Llama-8B	`instruction-023-l1024`	FG	no	0.624	0.855
R1-Llama-8B	`continuation-023-l1024`	FG	no	0.630	0.749
PatternKV K4V2, 1024 (H64/H64): geometric mean			8/8 func.	**0.515**	**0.729**
Qwen3-4B-Thinking	`code-074-l1024`	Ind.	yes	1.000	1.000
Qwen3-4B-Thinking	`continuation-074-l1024`	MHG	no	0.942	0.972
Qwen3-4B-Thinking	`instruction-074-l1024`	FG	no	0.979	0.956
Qwen3-4B-Thinking	`math-074-l1024`	FG	no	0.840	0.921
OSCAR rotation/clip INT2, 1024 (H64/H64): geometric mean			3/4 func.	**0.938**	**0.962**

Table 5: Wrapped-base rows of Table 1: KVarN (source-011, 4B/8B and the 14B column), PatternKV (source-023, H64/H64) and the OSCAR rotation/clip codec (source-074, H64/H64, single supported checkpoint; sink=0, recent=0).

D OUTCOME CONFIRMATIONS S1 AND S2

S1 contains 74 fresh MATH-500 and HumanEval identities; S2 contains 100 fresh math identities. Each source is evaluated on 2 models. S1 uses prefix retention as its primary outcome, while S2 uses task accuracy. Tables 6 and 7 report source-level and per-model results.

Stratum	n	KVarN	TRACEKV	Δ	95% CI	W/L	p
S1 (source-126): 148 cells							
all	148	32.2	46.6	+14.4	[+5.3, +25.3]	66/28	$1.1\mathrm{e}{-4}$
Qwen3-4B	74	28.1	38.1	+10.0	[-4.1, +28.5]	29/13	0.020
R1-Llama-8B	74	36.4	55.2	+18.8	[+8.1, +32.6]	37/15	0.003
S2 (source-128): 200 cells							
all	200	37.5	60.3	+22.8	[+10.8, +38.0]	81/41	$3.7\mathrm{e}{-4}$
Qwen3-4B	100	39.9	62.2	+22.2	[+8.5, +40.4]	39/21	0.027
R1-Llama-8B	100	35.1	58.5	+23.3	[+5.3, +49.7]	42/20	0.007

Table 6: Free-running prefix retention R_{1024} (mean tokens retained; Δ = TRACEKV+KVarN minus KVarN; paired-bootstrap 95% intervals from the frozen summaries). Retention is right-censored at $h{=}1024$ when both runs end identically.

Stratum	n	BF16	KVarN	TRACEKV	Δ (pp)	95% CI (pp)	W/L	p
S1 (source-126): 148 cells								
all	148	64.9	54.1	63.5	+9.5	[+0.7, +17.6]	29/15	0.049
Qwen3-4B	74	60.8	48.6	59.5	+10.8	–	16/8	0.152
R1-Llama-8B	74	68.9	59.5	67.6	+8.1	–	13/7	0.263
S2 (source-128): 200 cells								
all	200	73.0	70.0	72.5	+2.5	[-4.0, +9.0]	23/18	0.533
Qwen3-4B	100	71.0	73.0	68.0	-5.0	–	11/16	0.442
R1-Llama-8B	100	75.0	67.0	77.0	+10.0	–	12/2	0.013

Table 7: Exact task accuracy (%) at the 4096-token budget; Δ is TRACEKV+KVarN minus KVarN in percentage points.

E MECHANISM ABLATIONS

The sign and path-geometry ablations (§4.4) reuse the 12 source-011 KVarN cells (the 8 4B/8B cells of Table 1's KVarN row plus the 4 14B cells), rerun on the newer hardware of Appendix G; every arm within a run shares the trace, gradients, candidate machinery and hardware, so contrasts are paired. Ratios are the selected cache's D_H relative to independent KVarN; "fb" marks a safety fallback to a baseline. The path-default confirmation (Table 10) is a separately frozen 9-cell test on fresh identities.

Model	Family	Signed sel.	D_{64}/Ind.	Magnitude sel.	D_{64}/Ind.	Mag./Signed
Qwen3-4B	math	Ind.	1.000	FG	0.935	0.935
Qwen3-4B	code	FG	1.907	FG	1.684	0.883
Qwen3-4B	instruction	MHG	0.592	Temp.	0.802	1.354
Qwen3-4B	continuation	Temp.	0.169	Temp.	0.169	1.000
R1-Llama-8B	math	FG	0.769	MHG	0.943	1.227
R1-Llama-8B	code	FG	0.764	FG	0.841	1.100
R1-Llama-8B	instruction	MHG	0.796	Ind.	1.000	1.257
R1-Llama-8B	continuation	FG	0.679	Temp.	1.143	1.682
R1-Qwen-14B	math	FG	0.921	MHG	1.105	1.200
R1-Qwen-14B	code	FG	0.526	Temp.	0.729	1.386
R1-Qwen-14B	instruction	MHG	1.242	MHG	1.115	0.898
R1-Qwen-14B	continuation	FG	0.795	Ind.	1.000	1.258
Geometric mean / fallbacks		2/12 fb	**0.744**	6/12 fb	**0.864**	**1.161**

Table 8: Sign ablation: the signed account (paper configuration) against a magnitude accumulator that enters per-step contributions as absolute values, H32 construction and selection, H64 evaluation. All cells use the source-011 1024-token prompts (`<family>-011-l1024`).

Model	Family	Grad. sel.	D_{64}/Ind.	Rand. sel.	D_{64}/Ind.	Temp. sel.	D_{64}/Ind.
Qwen3-4B	math	Ind.	1.000	Ind.	1.000	MHG	0.756
Qwen3-4B	code	FG	1.907	MHG	1.095	MHG	0.478
Qwen3-4B	instruction	MHG	0.592	MHG	0.643	FG	0.585
Qwen3-4B	continuation	Temp.	0.169	Temp.	0.169	Temp.	0.169
R1-Llama-8B	math	FG	0.769	Ind.	1.000	FG	0.828
R1-Llama-8B	code	FG	0.764	MHG	1.098	FG	0.860
R1-Llama-8B	instruction	MHG	0.796	Ind.	1.000	FG	0.831
R1-Llama-8B	continuation	FG	0.679	Temp.	1.143	FG	0.760
R1-Qwen-14B	math	FG	0.921	MHG	0.609	MHG	0.428
R1-Qwen-14B	code	FG	0.526	FG	0.414	FG	0.378
R1-Qwen-14B	instruction	MHG	1.242	MHG	0.951	FG	1.269
R1-Qwen-14B	continuation	FG	0.795	MHG	1.429	MHG	0.848
Geometric mean / fallbacks		2/12 fb	**0.744**	5/12 fb	**0.780**	1/12 fb	**0.613**

Table 9: Path-geometry ablation on the source-011 prompts: gradient-space block path (default), seeded random within-block order, and plain temporal order, all with the signed account.

Model	Family	Grad.	D_{32}	D_{64}	Rand.	D_{32}	D_{64}	Temp.	D_{32}	D_{64}
Qwen3-4B	math	MHG	0.641	0.699	MHG	0.852	1.000	FG	0.653	0.726
Qwen3-4B	instruction	FG	0.570	0.658	MHG	0.630	0.813	MHG	0.541	0.617
Qwen3-4B	continuation	FG	0.529	0.767	Temp. (fb)	0.616	0.646	MHG	0.519	0.683
R1-Llama-8B	math	MHG	0.309	0.547	MHG	0.508	0.593	FG	0.320	0.676
R1-Llama-8B	instruction	MHG	0.685	0.734	Ind. (fb)	1.000	1.000	MHG	0.827	0.895
R1-Llama-8B	continuation	FG	0.898	0.824	Ind. (fb)	1.000	1.000	MHG	0.863	0.746
R1-Qwen-14B	math	Temp. (fb)	0.851	0.865	Temp. (fb)	0.851	0.865	MHG	0.594	0.620
R1-Qwen-14B	instruction	MHG	0.525	0.922	Ind. (fb)	1.000	1.000	FG	0.633	1.037
R1-Qwen-14B	continuation	MHG	0.630	0.650	MHG	0.747	1.143	FG	0.573	1.021
Geometric mean (fallbacks)		1/9 fb	**0.602**	**0.732**	5/9 fb	**0.780**	**0.877**	0/9 fb	**0.593**	**0.766**

Table 10: Frozen path-order comparison on 9 fresh prompt–model pairs (`math-264-l1024`, `instruction-012-l1024`, `continuation-046-l1024`). Columns give the selected candidate and its D_{32}/D_{64} relative to independent KVarN under gradient-path, random-order, and temporal-order processing.

F EFFECTIVE-BIT ACCOUNTING

Table 11 derives the persistent rate of each wrapped representation from the stored code and metadata counts recorded in the frozen result files (one 1024-token Qwen3-4B cell per representation; every candidate within a cell records identical counts, and the recorded per-element rate is identical across all cells of a row). Rates are bits per cached element (key and value elements counted together).

Representation	Codes	Metadata scalars	Meta. bits	Indices / mask	Derived	Recorded
Affine (K4V2, G64)	3.000	2 per 64 (scale, offset)	–	–	–	3.000 + metadata
KVarN (K4V2, G128)	3.000	3 per 128 (FP16)	0.375	–	3.375	3.375
PatternKV (K4V2, G128, 32 bases)	3.000	6 per 128 (FP16)	0.75	0.03906 / 0.003906	3.793	3.793
OSCAR rotation/clip codec (INT2, G128)	2.000	2 per 128 (FP32)	0.5	–	2.5	2.5

Table 11: Per-element persistent storage. "Codes" averages 4-bit keys and 2-bit values; "Metadata scalars" are the per-group scale/zero (and, for KVarN, dual-axis normalization) scalars with their stored width; "Indices / mask" are PatternKV's packed centre indices ($\lceil \log_2 32 \rceil = 5$ bits per token per stream) and its one-bit value residual mask. The affine base stores two scalars per 64-element group whose width the frozen protocols match but do not pin, so no single rate is quoted for it in the main text. The OSCAR rotation/clip codec uses sink=0 and recent=0.

Derivations. KVarN: $3 + 3 \cdot 16/128 = 3.375$ bits; each 128-token Sinkhorn tile stores two per-line scalars (quantization scale and zero) and one per-line dual-axis scale in FP16, i.e. 3 FP16 scalars per 128 elements. PatternKV: $3 + 6 \cdot 16/128 + 0.03906 + 0.003906 = 3.793$ bits: FP16 per-head centres and affine min/scale metadata, 5-bit centre assignments per token (32 bases), and a one-bit value mask. OSCAR rotation/clip codec: $2 + 2 \cdot 32/128 = 2.50$ bits: INT2 codes for keys and values with FP32 scale and zero per 128-element group, with sink=0 and recent=0; its 128×128 rotations are model-level parameters and are not cache storage. In every case the controller changes which codes and per-group scalar *values* are stored, never their number, width, or layout; the shared path is a deterministic construction order and is not stored.

G ENVIRONMENT AND MODEL REVISIONS

Environment. Table 12 lists the hardware and software environments used for the frozen evaluations.

Host	GPUs	Software	Used for
Host A	2x NVIDIA RTX PRO 5000 Blackwell	torch 2.9.0+cu128; transformers 4.57.6; Python 3.12.13; CUDA 12.8	Frozen representation and trajectory evaluations
Host B	8x RTX 6000D, SM120	torch 2.9.0+cu128; transformers 4.57.6; Python 3.12	Longer-prompt, 14B, and S1 evaluations
Host C	8x RTX PRO 6000D Blackwell	not recorded	Sign ablation (2026-08-25); S2 (source-128, frozen 2026-08-25; host inferred from the run location, not recorded in its files)
Host D	8x RTX PRO 6000D	not recorded	Path-geometry ablation (2026-09-02)
Host E	8x RTX PRO 6000D	torch 2.11.0+cu128; transformers 4.57.6; datasets 5.0.1	Path-default confirmation (2026-09-02); sign and SQuat composition studies (2026-09-09)

Table 12: Recorded execution environments. Software versions are listed where the run set recorded them; the Hosts B–E share the same virtual-environment lineage (transformers 4.57.6).

Model and base-code revisions. Table 13 lists the Hugging Face revisions pinned by the frozen protocols and stamped into every per-cell record (verified identical across all records in the artifact), and the commits of the public KVarN, PatternKV and OSCAR implementations wrapped.

Object	Revision / commit	Role
Qwen/Qwen3-4B	`1cfa9a72089121264592 14e8b04321603b3df60c`	HF revision
deepseek-ai/DeepSeek-R1-Distill-Llama-8B	`6a6f4aa4197940add577 24a7707d069478df56b1`	HF revision
deepseek-ai/DeepSeek-R1-Distill-Qwen-14B	`1df8507178afcc1bef68 cd8c393f61a886323761`	HF revision
Qwen/Qwen3-4B-Thinking-2507	`768f209d9ea81521153e d38c47d515654e938aea`	HF revision
KVarN reference code	`7586257f1c632e63187b facbbe21ccb51540f7b3`	git commit
PatternKV reference code	`aba09a82e14732f6a0ed 1f2b133925ff368d0538`	git commit
OSCAR reference code	`41ebcdba3db5f0ce1339 c3727caea80df575d437`	git commit
OSCAR rotation zoo	`b645aa31bc8a9d0fa572 dfbb95ad60fc35ac8f6f`	git commit

Table 13: Pinned model revisions (Hugging Face) and wrapped-base code commits, as recorded in the frozen protocols and in every per-cell record.

H ADDITIONAL MECHANISM RESULTS

The sign ablation is repeated on a fresh source, and the value controller is evaluated in combination with SQuat's key quantizer. Both studies use the same protocol across the compared arms; per-cell results are reported in Tables 14 and 16–17.

Model	Prompt	Signed sel.	D_{64}/Ind.	Magnitude sel.	D_{64}/Ind.	Mag./Signed
Qwen3-4B	`math-264-l1024`	MHG	0.699	MHG	0.863	1.234
Qwen3-4B	`instruction-012-l1024`	FG	0.658	Temp.	0.791	1.203
Qwen3-4B	`continuation-046-l1024`	FG	0.767	FG	0.630	0.822
R1-Llama-8B	`math-264-l1024`	MHG	0.547	Temp.	0.827	1.512
R1-Llama-8B	`instruction-012-l1024`	MHG	0.734	FG	0.700	0.954
R1-Llama-8B	`continuation-046-l1024`	FG	0.824	MHG	0.877	1.065
R1-Qwen-14B	`math-264-l1024`	Temp.	0.865	MHG	1.015	1.173
R1-Qwen-14B	`instruction-012-l1024`	MHG	0.922	MHG	1.103	1.196
R1-Qwen-14B	`continuation-046-l1024`	MHG	0.650	FG	0.954	1.468
Geometric mean / fallbacks (fresh source, 9 cells)		1/9 fb	**0.732**	2/9 fb	**0.851**	**1.162**
Source-011 diagnostic (Table 8, 12 cells)		2/12 fb	0.744	6/12 fb	0.864	1.161

Table 14: Fresh-source sign ablation on 9 prompt–model pairs over three models; H32 construction and selection, D_{64} evaluation. Columns show the selected candidate and its distortion relative to independent KVarN; "fb" marks selection of a baseline candidate.

	Fresh source (9 cells)			Source-011 (12 cells)		
Horizon	Signed	Magnitude	Mag./Signed	Signed	Magnitude	Mag./Signed
D_1	0.539	0.695	1.290	0.550	0.656	1.193
D_8	0.544	0.674	1.238	0.574	0.719	1.254
D_{16}	0.579	0.739	1.277	0.481	0.593	1.233
D_{32} (construction, selection)	0.602	0.755	1.254	0.499	0.655	1.312
D_{64} (evaluation)	0.732	0.851	1.162	0.744	0.864	1.161

Table 15: Geometric-mean selected/independent distortion at each horizon for the signed and magnitude accumulators, on the fresh source (9 cells) and source-011 diagnostic (12 cells).

I COMPARISON WITH FIXED FIRST-ORDER DESCENT

To isolate the ledger rule, we compare signed error balancing with a greedy first-order control while keeping the self-generated trace, gradients, processing path, codec, feedback gain, and candidate-generation procedure fixed. For independent reconstruction q_i^I, the control accumulates the signed

Key bits	Subspace dim	SQuat/KVarN keys	Subspace term	Gain, KVarN keys	Gain, SQuat keys	Δ	Fallbacks
2	5	0.812	0.920	0.866	0.842	-0.024	2/1/2
2	20	0.810	0.918	0.866	0.861	-0.006	2/3/2
2	60	0.725	0.822	0.866	0.831	-0.036	2/1/2
4	5	0.964	1.025	0.732	0.756	+0.024	1/0/2
4	20	0.978	1.040	0.732	0.713	-0.019	1/1/2
4	60	0.954	1.015	0.732	0.787	+0.055	1/1/2

Table 16: TraceKV combined with SQuat's key quantizer on the same 9 fresh prompt–model pairs across a key-width and subspace-dimension grid. The table compares key-side distortion and TraceKV's value-side gain under KVarN and SQuat keys; lower distortion ratios indicate improvement. The largest difference between the two value-side gains is 0.055.

Key configuration	Gain D_{32}	Gain D_{64}	Observed max D_{64}	Fallbacks	Key NMSE	Keys vs. KVarN
KVarN keys (paper baseline)	0.602	0.732	0.922	1/9	0.00314	–
SQuat keys (λ=0.001)	0.611	0.713	1.000	1/9	0.00196	0.978
SQuat base keys (λ=0)	0.704	0.788	0.979	2/9	0.00160	0.940

Table 17: Frozen 4-bit composition comparison on 9 fresh prompt–model pairs (K4V2, subspace dimension 20). Columns show TraceKV's selected-cache distortion at D_{32} and D_{64} under KVarN keys, SQuat keys, and the SQuat base quantizer with its subspace term disabled.

linearized change

$$a_t^{(m,b)} = a_{t-1}^{(m)} + \langle \bar{g}_{\pi_t}^{(m)}, q_{\pi_t}^{(b)} - q_{\pi_t}^{I} \rangle, \qquad b_t = \arg \min_{b \in \{0,1\}} \max_m a_t^{(m,b)}. \tag{12}$$

It chooses the feedback action that minimizes this fixed-gradient proxy; the signed controller instead minimizes the magnitude of its accumulated projection. We evaluate both rules on 64 frozen 1024-token inputs across two models (128 prompt–model pairs), with the same replay-based selector and independent and temporal-feedback baselines.

The signed single-objective candidate has a geometric-mean construction KL ratio of 0.606 relative to independent quantization; the matched linear candidate has ratio 8.105. For the two-objective variants, the corresponding ratios are 0.624 and 4.242. Replay selection rejects the linear candidates in all 128 pairs. This controlled comparison shows that minimizing the fixed signed linear proxy does not by itself produce better reconstructed caches in this setting. The control uses no line search, so the comparison is specific to this fixed-gradient procedure.