

Research Article

Beyond Perplexity: The Quality–Memory Frontier of Native Mixed-Precision KV Caching in llama.cpp

Petros Fanioudakis¹¹. Independent researcher

The key–value (KV) cache is a growing memory bottleneck for long-context autoregressive inference, but low-bit cache policies are frequently selected from perplexity alone. We implement native packed mixed-precision KV caching in `llama.cpp`, including Q3 formats, a four-level non-uniform Q2 value codebook, direct quantized cache writes, and mixed-format CUDA FlashAttention. A direct key-precision ablation on Qwen3–4B shows that Q3 keys increase held-out perplexity by approximately 20% and Q2 keys are catastrophic, whereas Q4 and Q8 keys have similar likelihood on the tested subsets; a 200-example GSM8K pilot nevertheless favors Q8 keys by 13 points, exposing a second likelihood–behavior mismatch. For values, the conservative Q4K–Q3V policy reduces KV storage by 13.89% and is statistically indistinguishable from Q4K–Q4V on full GSM8K, HumanEval, and MBPP. The aggressive Q4K–MixV–21 policy reduces KV storage by 20.99% and slightly lowers held-out perplexity, yet its strict GSM8K accuracy falls by 2.81 points (paired 95% CI $[-5.53, -0.08]$). At 32K populated context, it saves 272 MiB of process VRAM. Occupancy-verified RULER experiments from 8K to 32K show task- and context-dependent changes rather than monotonic degradation. Offline sensitivity analysis further shows that the fixed NU–Q2 codebook is within 1.1% of the deployment-data MSE optimum and reduces reconstruction MSE by 73.7% relative to uniform Q2. These results support a two-point deployment frontier and demonstrate that perplexity-preserving KV compression is not necessarily behavior-preserving.

Corresponding author: Petros Fanioudakis, pfanioudakis@gmail.com

1. Introduction

Autoregressive transformers [1] retain a key–value (KV) cache to avoid recomputing prior attention states. For L layers, H_{kv} KV heads, head dimension d , context length T , and effective storage costs b_K and b_V in bytes per element, the cache requires

$$B_{KV}(T) = TLH_{kv}d(b_K + b_V). \quad (1)$$

The linear dependence on T makes cache memory increasingly important for long contexts, concurrency, and consumer GPUs. Grouped-query attention reduces H_{kv} [2], paged allocation improves utilization [3], and IO-aware kernels reduce attention traffic [4]; numerical representation remains a direct lever on memory footprint and bandwidth.

KV-cache quantization has developed rapidly. KIVI introduced asymmetric key/value treatment [5]; KVQuant combined non-uniform datatypes, outlier handling, and pre-RoPE key quantization [6]; QAQ studied quality-adaptive precision [7], and later work introduced rotations, sliding windows, layer-wise allocation, query-aware precision, and rate–distortion optimization [8][9][10][11][12][13]. KVarN further shows that low-bit errors can accumulate during reasoning-oriented decoding [14]. These methods demonstrate that aggressive KV compression is possible, while also making comparisons difficult: published systems differ in model family, bit accounting, calibration, quantization axis, outlier storage, kernels, and benchmark harness.

This work focuses on a deployment-oriented question: what quality–memory frontier is obtained when mixed cache types are implemented natively inside `llama.cpp` [15] and evaluated beyond perplexity on an 8-GB consumer GPU? We do not claim numerical superiority over published methods that were not rerun under the same model and runtime. Instead, we contribute:

- native packed Q3 and non-uniform Q2 value formats across the relevant GGML, CPU, cache–write, and CUDA attention paths;
- a direct key-precision ablation showing that Q4 is the practical key floor under the tested scalar formats, while Q8 can retain more reasoning behavior at much higher storage cost;
- a conservative 13.89%–saving value policy that is statistically indistinguishable from Q4 values on the tested downstream tasks, and an aggressive 20.99%–saving policy that preserves perplexity but reduces mathematical reasoning;

- calibration, long-context uncertainty, and correctness analyses that expose several initially invalid experiment paths rather than hiding them.

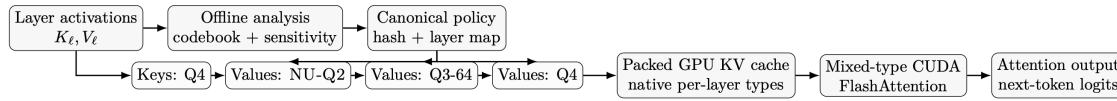


Figure 1. System overview. Policy selection is offline. At inference, each layer writes directly into its packed cache representation and the selected CUDA attention path consumes mixed K/V types without materializing a full FP16 cache.

2. Related Work and Scope

Asymmetric quantization.

Keys determine attention scores, whereas values are aggregated after the softmax. Prior work reports different outlier structure and preferred axes for the two tensors [5][6][7]. Our direct ablation in section 5.1 confirms strong asymmetry for the tested Qwen3-4B runtime: reducing keys from Q4 to Q3-64 increases perplexity by roughly 20%, while reducing values from Q4 to Q3-64 does not measurably reduce downstream accuracy.

Mixed precision and calibration.

KVTuner searches layer-wise key/value pairs [10], KVmix uses gradient-based layer importance [11], MixKVQ includes query relevance [12], and RateQuant calibrates quantizer-specific distortion curves before rate allocation [13]. Our policy is deliberately coarser: values are assigned one of three native packed formats per layer. This sacrifices allocation flexibility in exchange for a simple static runtime layout and direct integration with `llama.cpp`.

Why no numerical head-to-head.

KIVI, KVQuant, QAO, MixKVQ, KVarN, and RateQuant use different base models, average-bit definitions, quantization granularity, transforms, outlier budgets, and serving stacks. Copying their reported scores into a common table would not create a controlled comparison, and a rushed reimplementaion would be

less credible than an explicit limitation. We therefore compare method properties in the discussion, but restrict numerical claims to policies run through the same model file, binaries, datasets, and harnesses.

Evaluation beyond perplexity.

RULER ^[16], GSM8K ^[17], HumanEval ^[18], MBPP ^[19], and TruthfulQA ^[20] complement held-out likelihood. LongBench ^[21] could not be executed in the available environment and is reported as missing evidence rather than replaced by a claim of equivalent coverage.

3. Design

3.1. Packed formats

Table 1 reports physical storage, including one FP16 scale per block. The custom NU-Q2 format uses the 64-element Q2 layout but changes the reproduction levels; Q3-64 reduces scale overhead relative to the 32-element Q3 block.

Format	Values	Payload	Total	Eff. bpv
Q2 / NU-Q2	64	16 B	18 B	2.25
Q3	32	12 B	14 B	3.50
Q3-64	64	24 B	26 B	3.25
Q4	32	16 B	18 B	4.50
Q8	32	32 B	34 B	8.50

Table 1. Packed cache formats. Payload excludes the FP16 scale.

For a block scale s and index $i \in \{0, 1, 2, 3\}$, NU-Q2 reconstructs

$$\begin{aligned} \hat{x} &= s c_i, \\ \mathbf{c} &= (-0.619326, -0.186363, \\ &\quad 0.186363, 0.619301). \end{aligned} \tag{2}$$

The codebook was derived from a modeled clipped Gaussian in block-normalized space, not fitted to captured deployment activations. The adjacent midpoint thresholds are approximately $(-0.402845, 0, 0.402832)$. Offline analysis on rotated deployment values later found the fixed levels to be within 1.1% of the runtime-MSE optimum and stable across calibration sizes and text/code windows (section 6.2).

3.2. Layer policies and byte accounting

For Qwen3-4B, $L = 36$, $H_{kv} = 8$, and $d = 128$, giving 1,024 key and 1,024 value elements per layer and token. Effective costs are 0.5625 B/element for Q4, 0.40625 for Q3-64, and 0.28125 for the 64-element Q2 layout. The resulting core policies are shown in table 2.

Policy	B/token	Relative	Saving
Q8K-Q8V	78,336	$1.889\times$	—
Q4K-Q4V	41,472	$1.000\times$	0.00%
Q4K-Q3V	35,712	$0.861\times$	13.89%
Q4K-MixV-21	32,768	$0.790\times$	20.99%
All-NU-Q2 V	31,104	$0.750\times$	25.00%
Uniform-Q2 V	31,104	$0.750\times$	25.00%

Table 2. Total K+V storage per token. “Relative” is versus Q4K-Q4V.

All paper policies keep keys at Q4 except explicit key-precision controls. Q4K-Q3V uses Q3-64 values in all layers. C2 protects value layers $\{0, 7, 20, 23\}$ at Q4, uses Q3-64 at $\{9, 14, 16, 34\}$, and assigns NU-Q2 to the remaining 28 layers.

3.3. Native execution path

The implementation adds type traits, CPU quantize/dequantize functions, CUDA dequantizers, vector-dot kernels, row write/read operations, and mixed $K\times V$ FlashAttention dispatch. Per-layer cache types are

validated before packed buffers are allocated and are written directly into the GPU cache, without a persistent full-precision mirror or a CPU/disk memory tier. The production CUDA paths are $Q4K \times Q4V$, $Q4K \times Q3-64V$, and $Q4K \times NU-Q2V$. Quantized values require FlashAttention, and unsupported combinations fail explicitly. A dispatch gap prevents $K=Q3-64$, $V=Q4$ from using the fused path, so that pair is used only in the controlled perplexity ablation with FlashAttention disabled.

4. Experimental Methodology

4.1. Models and hardware

The primary model is Qwen3-4B-Instruct with Q4_K_M weights [22]. Same-family evidence uses Qwen3-1.7B, and cross-family transfer uses Llama-3.2-1B-Instruct. All primary experiments ran on one NVIDIA RTX 3070 Laptop GPU with 8 GB VRAM. The environment records CUDA toolkit 13.3, driver 610.43.03, CMake 4.4.0, Python 3.13, and an Intel i7-12700H. No second GPU, multi-GPU configuration, or non-CUDA backend was tested.

4.2. Data separation and likelihood evaluation

Calibration, policy selection, and final evaluation use disjoint files. Final WikiText-derived [23] and OpenWebText-derived [24] test files were not used to define the codebook or layer map. The primary frontier uses context 4,096 and reports aggregate PPL over 149,431 and 542,455 scored tokens. A separate paired OpenWebText chunk analysis is used only for its paired confidence interval. The key-precision ablation uses a fixed first-64-chunk subset at context 2,047 and is not numerically mixed with the primary PPL table.

4.3. Downstream and long-context evaluation

GSM8K uses strict first-attempt exact match over 1,319 examples. HumanEval and MBPP use deterministic pass@1 over 164 and 500 examples. TruthfulQA reports mean bleu_max over 817 examples and is treated as supporting evidence.

The final RULER matrix contains four policies, five tasks, and 8K, 16K, and 32K contexts with 100 examples per cell. No cell has a null output, truncation, or prompt-occupancy violation. RULER evaluates task quality through 32K. The systems evaluation reports populated-context behavior through 32K. Analytical cache accounting is also reported at 64K, while task quality and decode throughput are not

claimed at that length. Native fractional scores are analyzed with paired bootstrap intervals. A secondary thresholded pass/fail analysis is reported separately where relevant and is never substituted for the native score.

4.4. Statistics

Shared benchmark examples are paired. We report exact two-sided McNemar tests for binary outcomes and paired percentile-bootstrap 95% confidence intervals for score differences. The central Q4K-Q3V family includes GSM8K, HumanEval, MBPP, and 15 RULER cells; Holm correction is applied within that declared family. Exploratory all-policy comparisons are not promoted to confirmatory claims.

5. Results

5.1. Keys are more sensitive than values

Tables 3 and 2 closes the design gap that previously motivated Q4 keys only qualitatively. With values fixed at Q4, Q3-64 keys increase PPL by 19.6% on the Wiki subset and 20.1% on the OWT subset. Q2 keys are catastrophic, reaching PPL 2,151.9 on the 16-chunk gate. Q8 and Q4 keys are close in subset PPL, but a 200-example GSM8K pilot favors Q8 keys by 13.0 points (paired CI [6, 20], McNemar $p = 0.0007$).

Key type	Wiki PPL	OWT PPL	GSM8K-200
Q8	12.3683	20.3589	77.5%
Q4	12.3227	20.3030	64.5%
Q3-64	14.7401	24.3900	not run
Q2	2,151.9 [†]	—	stopped

Table 3. Key-precision ablation with Q4 values fixed. PPL uses 64 chunks at context 2,047; Q2 is a 16-chunk stop gate.

[†] Sixteen Wiki chunks; $174.6 \times Q4$ PPL. Q3-key GSM8K was not run because the fused-attention dispatch lacks $K=Q3-64$, $V=Q4$.

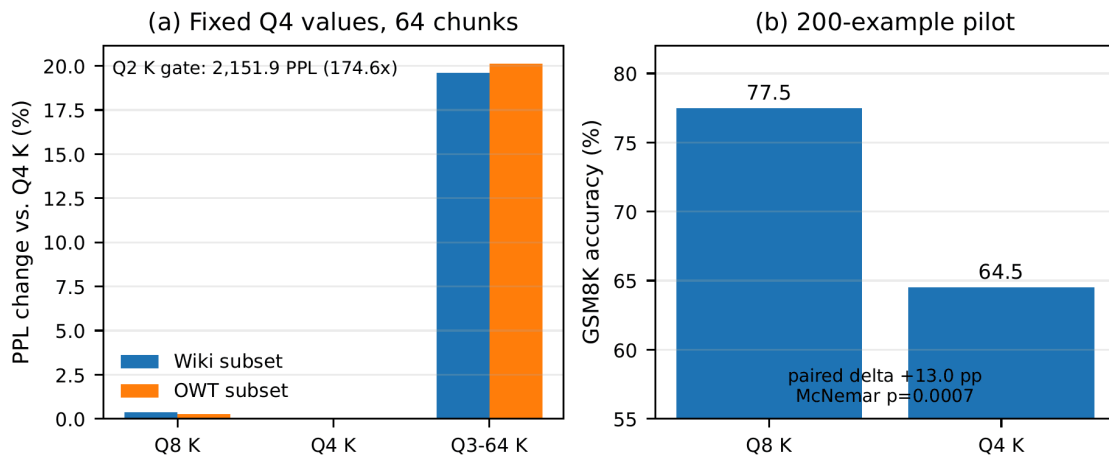


Figure 2. Q4 is the practical key floor under the tested scalar formats, but Q8 keys retain substantially more GSM8K accuracy in the bounded pilot. Similar PPL is therefore not evidence of behavioral equivalence even on the key axis.

The scoped conclusion is not that four bits are universally sufficient. It is that Q3 and Q2 keys are unsuitable for the tested memory-efficient scalar path, while Q8 represents a larger quality-oriented reference.

5.2. The main frontier separates likelihood from reasoning

Tables 4 and 3 presents the primary Qwen3-4B policies. Q4K-Q3V saves 13.89% and changes Wiki/OWT PPL by only +0.43%/ +0.33%. C2 saves 20.99% and slightly improves both aggregate PPL values. All-layer non-uniform Q2 is materially better than uniform Q2 at identical storage, but it still degrades downstream quality.

Cache policy	KV B/token	Save	Wiki PPL	OWT PPL	GSM8K	HumanEval	MBPP	TQA BLEU
Q8K-Q8V	78,336	–	12.2680	18.5222	75.21	76.83	53.80	13.73
Q4K-Q4V	41,472	0.00%	12.2919	18.4667	65.43	71.34	51.80	11.39
Q4K-Q3V	35,712	13.89%	12.3452	18.5276	67.10	73.17	52.60	–
Q4K-MixV-21	32,768	20.99%	12.2435	18.3585	62.62	71.95	50.60	11.94
All-NU-Q2 V	31,104	25.00%	12.5368	18.7619	58.53	65.85	48.00	11.12
Uniform-Q2 V	31,104	25.00%	15.1708	23.5082	31.01	65.85	30.80	7.79

Table 4. Primary Qwen3-4B quality-memory frontier. All rows use the same Q4_K_M weight model. Task scores are percentages.

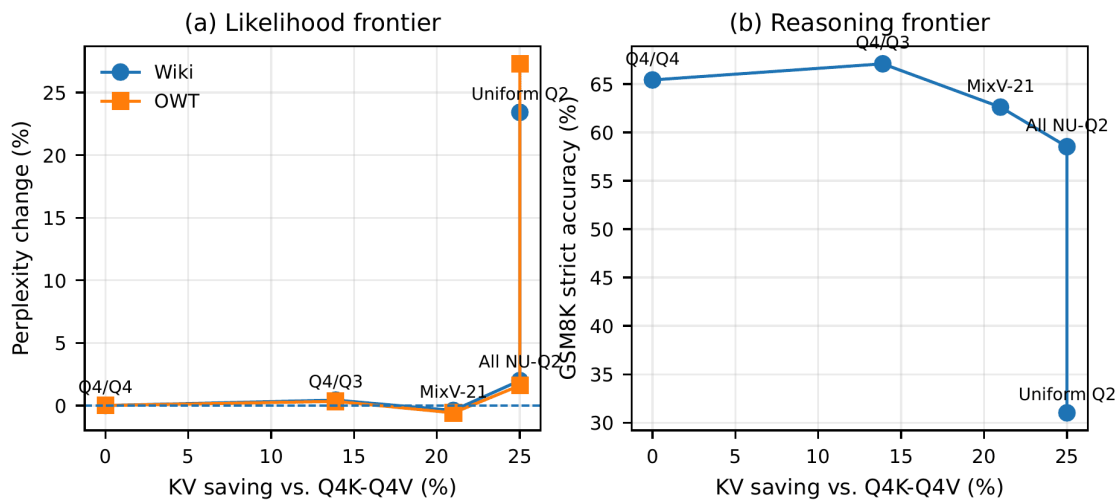


Figure 3. The likelihood and reasoning frontiers disagree. C2 is favorable under PPL but worse on GSM8K, while uniform Q2 collapses both.

The paired downstream comparison in table 5 supports conservative wording for Q4K-Q3V. Its positive point estimates on GSM8K, HumanEval, and MBPP are statistically unresolved. We therefore call it quality-preserving in the tested samples, not quality-improving.

Task	Δ (pp)	95% CI	McNemar p
GSM8K	+1.67	[−1.06, 4.32]	0.249
HumanEval	+1.83	[−4.88, 8.54]	0.728
MBPP	+0.80	[−2.80, 4.40]	0.741

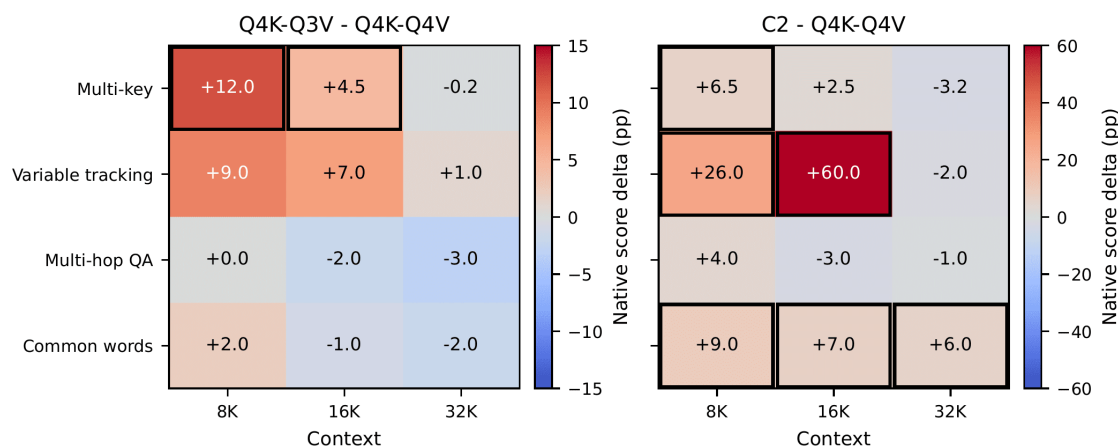
Table 5. Paired Q4K-Q3V minus Q4K-Q4V differences. Confidence intervals are paired bootstrap intervals.

C2 demonstrates the central mismatch. Strict GSM8K falls by 2.81 points relative to Q4K-Q4V; the paired CI is [−5.53, −0.08] and exact McNemar $p = 0.047$. This is a borderline pre-specified result and would not survive an indiscriminate correction over every benchmark cell, but it is directionally consistent with the larger Q8-to-Q4 key pilot and with recent work on error accumulation ^[14]. HumanEval changes by +0.61 point and MBPP by −1.20 points, with both intervals including zero.

5.3. Long-context effects are task- and length-dependent

Figure 4 replaces the dense five-panel line plot with native-score deltas and paired uncertainty. Q4K-Q3V improves multi-key retrieval by 12.0 points at 8K (CI [8.0, 16.0]) and 4.5 points at 16K (CI [0.25, 8.5]); the latter effect is similar in magnitude to the empirical self-resampling noise floor and should be treated cautiously. At 32K its multi-key difference is −0.25 point with an interval spanning zero.

C2 increases native variable-tracking score by 26 and 60 points at 8K and 16K, while its 32K difference is −2 points. Its native multi-key deltas are +6.5, +2.5, and −3.25 points. The 32K native fractional interval includes zero, although a separate thresholded pass/fail analysis detects a significant reversal ($p = 0.0013$). These are distinct metrics and are not presented interchangeably.



Black border: paired 95% bootstrap interval excludes zero. Single-key retrieval omitted (99-100% throughout).

Figure 4. Native RULER score deltas versus Q4K-Q4V (100 paired examples per cell). Borders indicate that the paired 95% bootstrap interval excludes zero; they do not imply survival of every multiple-comparison correction. The response to quantization is non-monotonic.

The large positive cells must not be interpreted as compression increasing intelligence. Deterministic quantization changes token trajectories, and finite benchmark cells can move in either direction. The defensible conclusion is that stable PPL does not imply stable long-context behavior.

5.4. Memory savings translate into process VRAM

The byte model predicts an 8,704 B/token gap between Q4K-Q4V and C2. Measured process-VRAM differences are 18, 34, 68, 136, and 272 MiB at 2K, 4K, 8K, 16K, and 32K populated contexts. At 32K, peak process VRAM falls from 4,188 to 3,916 MiB. The 6.5% process-level reduction is smaller than the 20.99% KV-only saving because weights and other buffers are unchanged.

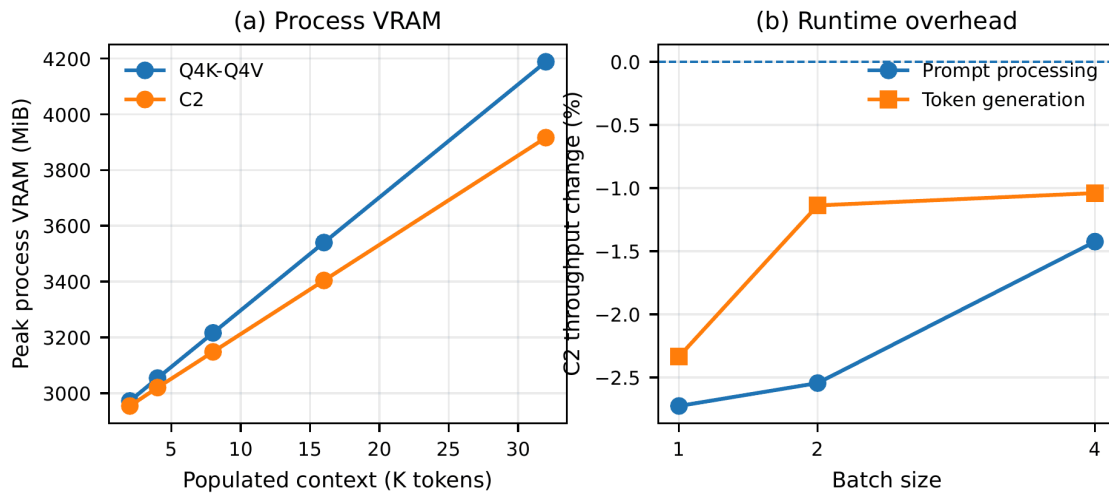


Figure 5. Systems results on the RTX 3070 Laptop. C2’s process-VRAM advantage scales linearly with populated context; prompt-processing and generation throughput decrease by roughly 1–3%.

Prompt-processing throughput is 2.7%, 2.5%, and 1.4% lower at batch sizes 1, 2, and 4. Generation throughput is 2.3%, 1.1%, and 1.0% lower. Two-slot latency rises by approximately 1.1%, and warm cache-reuse latency changes from 0.664 to 0.674 s.

6. Ablations and Transfer

6.1. The non-uniform codebook is load-bearing

At identical all-Q2 storage, uniform Q2 produces Wiki/OWT PPL 15.1708/23.5082, whereas NU-Q2 produces 12.5368/18.7619. The end-to-end difference is supported by separate binary hashes and distinct output fingerprints, avoiding an earlier phantom-control failure in which both settings used the same kernel.

6.2. Calibration robustness

The fixed codebook was designed from a clipped Gaussian rather than captured deployment values. To test fragility, eight post-rotation value windows (six text, two code) were refit using whole quantization blocks and the runtime scale convention. Across 12 size/seed fits, maximum level drift is 0.0371 in normalized units. The fixed codebook’s MSE is 1.008–1.011 times the fitted optimum, and text/code fits

select the same rounded levels. On real deployment captures, NU-Q2 reduces reconstruction MSE by 73.7% relative to uniform Q2; the larger 96% figure applies only to the design Gaussian.

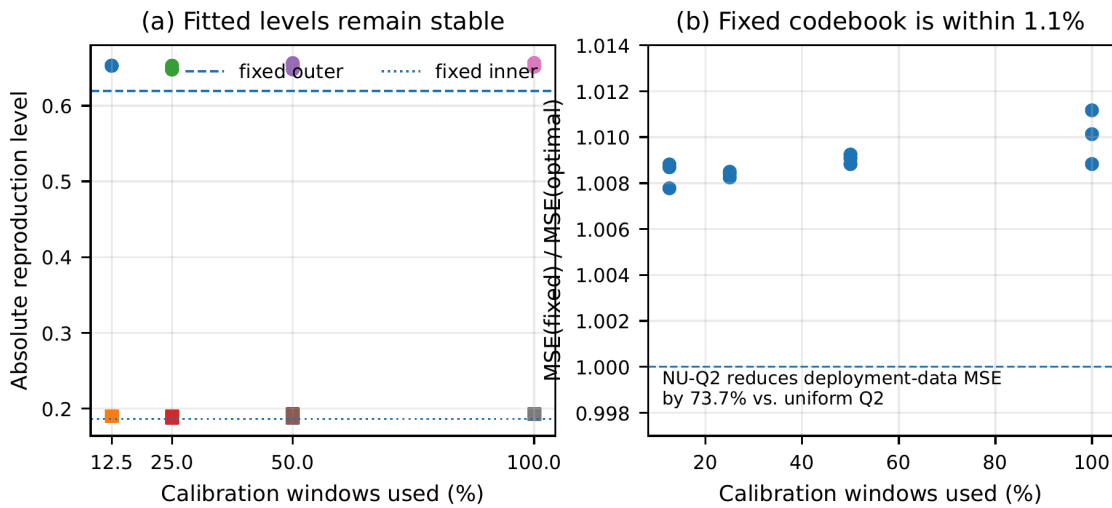


Figure 6. Calibration sensitivity. The fitted levels are stable from 12.5% to 100% of the available windows, and the fixed codebook remains within 1.1% of the runtime-MSE optimum.

6.3. Layer selection is useful, not optimal

At the 21% budget, C2 beats 30 of 32 random equal-memory maps on the selection PPL, but random-23 (exploratory/post-hoc) performs better. The heuristic is therefore informative but not globally optimal, and exact layer placement remains a search problem. The main conclusions depend more strongly on the precision budget and codebook choice than on claiming an optimal map.

6.4. Transfer does not establish generality

On Qwen3-1.7B, a transferred C2-style policy lowers paired Wiki PPL from 18.4777 to 16.6803 and OWT PPL from 23.4825 to 22.3532, but GSM8K falls from 65.43% to 62.02%. This repeats the likelihood-behavior mismatch at smaller scale. On Llama-3.2-1B, Qwen-derived and heuristic maps remain worse than the Q4 baseline; no Llama-specific Lloyd-Max codebook was fitted. These experiments are transfer probes, not evidence of broad family generalization.

7. Correctness, Failures, and Reproducibility

Validation includes packing round trips, CPU/CUDA parity, row writes, mixed-attention checks, backend-operation tests, end-to-end inference, and Compute Sanitizer. The reported Q4K–Q3V, C2, and all-NU-Q2 value paths complete without selected-path test or sanitizer failures. Invalid policy configurations terminate explicitly rather than falling back silently.

A central methodological contribution is the explicit failure ledger in table 6. Several initially plausible results were invalid and were replaced rather than rationalized.

Issue	Symptom	Fix / verification	Impact on evidence
Q2 packed-index bug	Potential out-of-bounds access for positions within a 64-value block	Corrected byte/bit indexing; dedicated boundary tests and sanitizer runs	Earlier affected kernels invalidated; final selected path re-run
Missing C2 runtime map	Mixed policy could silently execute as uniform Q4 values	Use an explicit validated per-layer map and verify the resolved runtime types	RULER matrix restarted before any C2 cell was accepted
Short “long-context” prompts	Nominal 8K–64K cells contained only hundreds of tokens	Rebuilt generators; enforced occupancy and truncation audits	Earlier results discarded; the 8K–32K matrix was regenerated
Code harness floor	HumanEval/MBPP near zero for every policy	Corrected chat tasks, stop strings, fence extraction, and execution harness	Full 164/500-example code results regenerated
Uniform-Q2 phantom	Uniform control silently reused non-uniform kernels	Separate binaries with distinct hashes and logit fingerprints	NU-vs-uniform ablation rerun end-to-end
Calibration metric errors	Offline refits reported large false gaps or degenerate levels	Use runtime scale, whole blocks, and global grid-refine optimizer	Calibration robustness results regenerated; fixed codebook within 1.1%
Q3-key FA dispatch gap	K=Q3-64, V=Q4 crashes in the fused path	Exclude that pair from fused downstream runs; use non-FA PPL only	Key ablation is scoped; no Q3-key GSM8K claim

Table 6. High-impact failures discovered during the study. “Impact” states how the paper evidence changed.

All reported results use the corrected implementation and validated cache policies.

8. Deployment Guidance

Table 7 turns the two-point frontier into an explicit decision rule.

Workload	Policy	Save	Guidance
Reason/code sensitive	Q4K-Q3V	13.9%	No resolved decline on the three tested tasks
Capacity constrained	Q4K-MixV-21	21.0%	PPL favorable; GSM8K and context-specific RULER risk
Maximum compression study	All-NU-Q2 V	25.0%	Better than uniform Q2; downstream loss remains
Avoid	Uniform-Q2 V	25.0%	Severe likelihood and downstream degradation

Table 7. Practical policy selection under the tested Qwen3-4B setting.

9. Limitations

- The primary frontier is Qwen3-4B on a single RTX 3070 Laptop GPU. No 7B/8B primary model, second GPU architecture, multi-GPU setup, or non-CUDA backend is evaluated.
- No published KV quantizer is reimplemented head-to-head. Cross-paper scores are not treated as controlled baselines because bit accounting, transforms, models, kernels, and harnesses differ.
- The key-precision PPL ablation uses 64 chunks at context 2,047, and its GSM8K result is a 200-example Q8/Q4 pilot. The Q3-key fused path remains unsupported.
- RULER task quality and measured decode performance are evaluated through 32K, and LongBench could not be executed. Analytical cache accounting extends to 64K, but no 64K task-quality claim is made.
- TruthfulQA uses bleu_max, and long free-generation degeneration is based on one prompt per task; both are supporting rather than headline evidence.
- The static codebook is not per-model data-optimal, although it is near-optimal on the tested deployment captures. Policy maps are static per layer, and random search found competitive maps.
- Qwen3-1.7B and Llama-3.2-1B are limited transfer probes. They do not establish cross-family or large-scale generality.
- Uniform Q2 is evaluated as a negative control but is not supported by the production CUDA path; the primary policies use Q4, Q3-64, and NU-Q2 cache types.

10. Conclusion

Native mixed-precision KV caching can reduce memory in a real consumer-GPU inference stack, but the safe precision depends on the behavior being protected. Keys are particularly sensitive: Q3 and Q2 keys damage likelihood severely, while Q8 keys retain more GSM8K behavior than Q4 in a bounded pilot. Values permit a useful frontier. Q4K-Q3V saves 13.89% and is statistically indistinguishable from Q4K-Q4V on the tested reasoning and code benchmarks. C2 saves 20.99% and slightly improves measured PPL, yet loses 2.81 GSM8K points and changes long-context task behavior non-monotonically. The practical lesson is direct: select cache precision with downstream and long-context evidence, not perplexity alone.

Appendix A. Full Policy Definitions

Policy	KV B/token	Saving	Value precision map
Q4K-Q4V	41,472	0.00%	Q4 at layers 0–35
Q4K-Q3V	35,712	13.89%	Q3–64 at layers 0–35
Q4K-MixV- 21	32,768	20.99%	Q4: 0,7,20,23; Q3-64: 9,14,16,34; NU-Q2: remaining 28 layers
Random-23	32,768	20.99%	Exploratory/post-hoc; Q4: 0,9,13,20; Q3-64: 3,4,5,6; NU-Q2: remaining 28 layers
All-NU-Q2	31,104	25.00%	NU-Q2 at layers 0–35
Uniform-Q2	31,104	25.00%	Uniform Q2 at layers 0–35

Table 8. Canonical primary policies for Qwen3-4B. Value-layer indices are zero-based.

Appendix B. Complete RULER Scores

Policy	Task	8K	16K	32K	Policy	Task	8K	16K	32K
Q4K-Q4V	Single key	100	100	100	Q4K-Q3V	Single key	100	100	100
	Multi-key	32.75	35.25	49.25		Multi-key	44.75	39.75	49.00
	Variable tracking	52	7	99		Variable tracking	61	14	100
	Multi-hop QA	19	16	23		Multi-hop QA	19	14	20
	Common words	79	87	92		Common words	81	86	90
C2	Single key	99	100	100	Random-23	Single key	99	100	100
	Multi-key	39.25	37.75	46.00		Multi-key	40.25	43.25	42.50
	Variable tracking	78	67	97		Variable tracking	70	57	100
	Multi-hop QA	23	13	22		Multi-hop QA	20	10	14
	Common words	88	94	98		Common words	87	92	97

Table 9. Occupancy-verified native RULER task scores (%). Each cell has 100 examples.

Appendix C. Supporting Transfer and Generation Evidence

On Qwen3-1.7B, the C2-style policy improves paired Wiki/OWT PPL but reduces GSM8K. On Llama-3.2-1B, all transferred or heuristic low-bit policies remain worse than Q4 values; no Llama-specific codebook was fitted. These results support the mismatch thesis but not a generalization claim.

Both Q4K-Q4V and C2 loop on a 4K-token greedy story prompt. C2 is worse on the single code-generation probe. Because there is one prompt per condition, these outputs are archived as failure cases rather than aggregate quality metrics.

Statements and Declarations

Data Availability

Code: github.com/PetrouilFan/llama.cpp

Use of Generative AI

Generative AI tools assisted with literature triage, code review, figure preparation, and language editing. The author verified the analyses and assumes full responsibility for the implementation, experiments, claims, and remaining errors. No AI system is listed as an author.

Acknowledgments

The author thanks the llama.cpp and GGML contributors for the open-source inference stack on which this work is built.

References

1. [△]Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017). "Attention Is All You Need." *NeurIPS*.
2. [△]Ainslie J, Lee-Thorp J, de Jong Michiel, Zemlyanskiy Y, Lebron F, Sanghai S (2023). "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints." *arXiv:2305.13245*.
3. [△]Kwon W, Li Z, Zhuang S, Sheng Y, Zheng L, Yu CH, Gonzalez JE, Zhang H, Stoica I (2023). "Efficient Memory Management for Large Language Model Serving with PagedAttention." *SOSP*.
4. [△]Dao T, Fu DY, Ermon S, Rudra A, Ré C (2022). "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness." *NeurIPS*.
5. [△]_a [△]Liu Z, Yuan J, Jin H, Zhou S, Ni J, Liu J, Hu X (2024). "KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache." *arXiv:2402.02750*.
6. [△]_a [△]Hooper C, Kim S, Mohammadzadeh M, Mahoney MW, Keutzer K, Gholami A (2024). "KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization." *arXiv:2401.18079*.
7. [△]_a [△]Dong S, Han Y, Zhang Y, Xu C, Sun J (2024). "QAQ: Quality Adaptive Quantization for LLM KV Cache." *arXiv:2403.04643*.

8. [△]Ashkboos S, Croci ML, Li G, Zhao M, Alistarh D, Hoefler T, De Sa C (2024). "QuaRot: Outlier-Free 4-Bit Inference in Rotated LLMs." arXiv:2404.00456.
9. [△]Dai Y, Shen Y, Zhao D, Zhang Y, Li J, Zhou S (2024). "SKVQ: Sliding-window Key and Value Cache Quantization for Large Language Models." arXiv:2405.06219.
10. [△][♭]Li Y, Zhang Y, Wang X, Chen Y, Jiang J (2025). "KV-Tuner: Sensitivity-Aware Layer-Wise Mixed-Precision KV Cache Quantization." arXiv:2502.04420.
11. [△][♭]Li F, Wang Y, Zhang Z, Liu Y, Yang M (2025). "KVMix: Gradient-Based Layer Importance-Aware Mixed-Precision Quantization for KV Cache." arXiv:2506.08018.
12. [△][♭]Zhang T, Zeng Z, Peng H, Zhuang H, Chen C (2026). "MixKVQ: Query-Aware Mixed-Precision KV Cache Quantization for Long-Context Reasoning." ACL. 7189–7204. doi:10.18653/v1/2026.acl-long.326.
13. [△][♭]Zuo F, Zhou Z, Cong H, Xi X, Leung HF (2026). "RateQuant: Optimal Mixed-Precision KV Cache Quantization via Rate-Distortion Theory." arXiv:2605.06675.
14. [△][♭]Muller LK, Bich P, Boretti C, Chang HM, Zhuang J, Cavigelli L (2026). "KVarN: Variance-Normalized KV-Cache Quantization Mitigates Error Accumulation in Reasoning Tasks." arXiv:2606.03458.
15. [△]ggml-org (2026). "llama.cpp: LLM Inference in C/C++." ggml-org. <https://github.com/ggml-org/llama.cpp>.
16. [△]Hsieh CP, Tseng WY, Chen CY, Lee SD, Lin CH, Lee HY (2024). "RULER: What's the Real Context Size of Your Long-Context Language Models?" arXiv:2404.06654.
17. [△]Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser L, Plappert M, Tworek J, Hilton J, Nakano R, Hesse C, Windle J, Lightman H, Chantry P, Jiang E, Beattie K, Ward C, Sastry J, Kerry P, Pradhan S, Klimov T, Agarwal R, Zhong A, Wu G, Jiang L, Lim J, Baid Y, Yuan J, Wang H, Mishkin P, Brundage M, Sastry G, Shulman M, Sutskever I, Zaremba W (2021). "Training Verifiers to Solve Math Word Problems." arXiv:2110.14168.
18. [△]Chen M, Tworek J, Jun H, Yuan Q, Pinto HPdeO, Kaplan J, Edwards H, Burda Y, Joseph N, Brockman G, Ray A, Puri R, Krueger G, Petrov M, Khlaaf H, Sastry G, Mishkin P, Chan B, Gray S, Ryder N, Mikhailov M, Debiak K, Hahn J, Bedford S, Sullivan S, Peterson J, Askell A, Mishchenko P, Pang T, Krueger F, Agarwal S, Sastry J, Kadavath S, Conerly D, de Oliveira Pinto HP, d'Abreu Azevedo G, Hernandez D, Chen B, Dachapally R, Luan J, Lim J, Campbell C, Huizinga J, Barnes T, Haas E, Shikhman S, Dowling J, Tang J, Plappert M, Chantzis S, Mousavi H, Jiang E, Beattie K, Ward C, Sastry J, Kerry P, Pradhan S, Klimov T, Agarwal R, Zhong A, Wu G, Jiang L, Lim J, Baid Y, Yuan J, Wang H, Mishkin P, Brundage M, Sastry G, Shulman M, Sutskever I, Zaremba W (2021). "Evaluating Large Language Models Trained on Code." arXiv:2107.03374.
19. [△]Austin J, Odena A, Nye M, Bosma MA, Henry H, Miranda C, Chen Y, Infante-Scher C, Miao H, Zhou S, Douglas P, Devlin J, Sutton C (2021). "Program Synthesis with Large Language Models." arXiv:2108.07732.

20. [△]Lin S, Hilton J, Evans O (2021). "TruthfulQA: Measuring How Models Mimic Human Falsehoods." *arXiv:2109.07958*.
21. [△]Bai Y, Lv X, Zhang J, Lyu H, Tang J, Huang L, Du Z, Liu X, Zeng A, Hou L, Dong Y, Tang J, Li J (2023). "LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding." *arXiv:2308.14508*.
22. [△]Yang A, Yang B, Hui B, Zheng B, Yu C, Li C, Liu C, Wang C, Zhou C, Li C, Zhu D, Zhang D, Li F, Ni G, Wang G, Fan H, Zhou H, Lin J, Zhang J, Pan J, An K, Lu L, Li M, Xue M, Ge N, Wang P, Zhang R, Ge S, Li S, Fan S, Wu S, Lu S, Shi T, Wang T, Tang T, Wang W, Yuan X, Xu X, Wang X, Zhang X, Pan Y, Yuan Y, Fan Y, Zhang Y, Yao B, Jiang J, Yang J, Zhou J, Chang S, Tan S, Bai J, Zhou J (2025). "Qwen3 Technical Report." *arXiv:2505.09388*.
23. [△]Merity S, Xiong C, Bradbury J, Socher R (2016). "Pointer Sentinel Mixture Models." *arXiv:1609.07843*.
24. [△]Gokaslan A, Cohen V (2019). "OpenWebText Corpus."

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.