RATEQUANT: ATTENTION-AWARE RATE–DISTORTION ALLOCATION FOR KV CACHE QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective KV cache compression combines accurate quantization with precision allocation to preserve model quality within a memory budget. Allocation is coupled through existing attention error and inputs from earlier quantized layers. We introduce **RATEQUANT**, a rate–distortion framework allocating KV precision across layers and heads through measured attention responses. These responses capture how adjacent bit-width changes reinforce or cancel the existing attention error. Dynamic programming combines them into budget-preserving joint updates, and remeasurement refreshes the responses for the next round. Full-model cross-entropy on separate calibration data then selects the final allocation. Our analysis connects normalized attention distortion to task loss under bounded suffix sensitivity and establishes exact optimization of the local additive objective. Across three models and two quantizers, RATEQUANT achieves the lowest perplexity in all 18 settings spanning 2–3-bit budgets, with reductions of 12.6–94.5% at 2 bits and 3.2–10.3% at 2.5 bits versus each setting's strongest tested baseline. Frozen allocations achieve the highest three-model mean scores in all 30 LongBench evaluation settings (five tasks, two quantizers, and three budgets), without task-specific recalibration.

1 INTRODUCTION

Serving a language model requires storing key and value states so that each new token can attend to the preceding context. The KV cache grows with sequence length and batch size, making it a substantial part of inference memory (Pope et al., 2023; Kwon et al., 2023). Quantizers such as KIVI, QuaRot, KVQuant, and TURBOQUANT reduce the precision of these states (Liu et al., 2024; Ashkboos et al., 2024; Hooper et al., 2024; Zandieh et al., 2026). Once a quantizer is chosen, a second decision remains: *where should a fixed precision budget be spent?* An additional bit can have different value across layers, heads, and the key and value components of attention.

Uniform precision leaves these differences unused. A global K/V split accounts for the different roles of keys and values, while recent methods allocate precision across layers or channels (Li et al., 2026; 2025; Liao & Wen, 2026; Xia et al., 2026). We study allocation jointly across layers, KV heads, and K/V components. Directly evaluating every candidate by full-network loss is expensive, because each change must be propagated through the remaining model. A local criterion should instead capture how a finite precision change affects the attention computation where it occurs.

Rate–distortion theory gives a natural formulation: minimize reconstruction error subject to a storage budget (Cover & Thomas, 2006). For KV allocation, the key question is which reconstruction error to measure. Quantizing keys changes attention weights; quantizing values changes the vectors those weights combine. The output projection mixes the contributions of different heads. Earlier allocation decisions also change the inputs seen by later layers. A component's allocation value therefore depends on the surrounding quantized computation, rather than only on the error in its own cached tensor.

We measure the attention output after projection, comparing quantized and unquantized attention at the same input from the current quantized prefix. If the existing attention residual is r and a precision change produces output difference δ, the squared error changes by $2\langle r, \delta\rangle + \|\delta\|^2$. Its value depends on whether the change reinforces or cancels the current residual. Two changes of the same magnitude can therefore have different allocation values, depending on their direction relative to the current

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

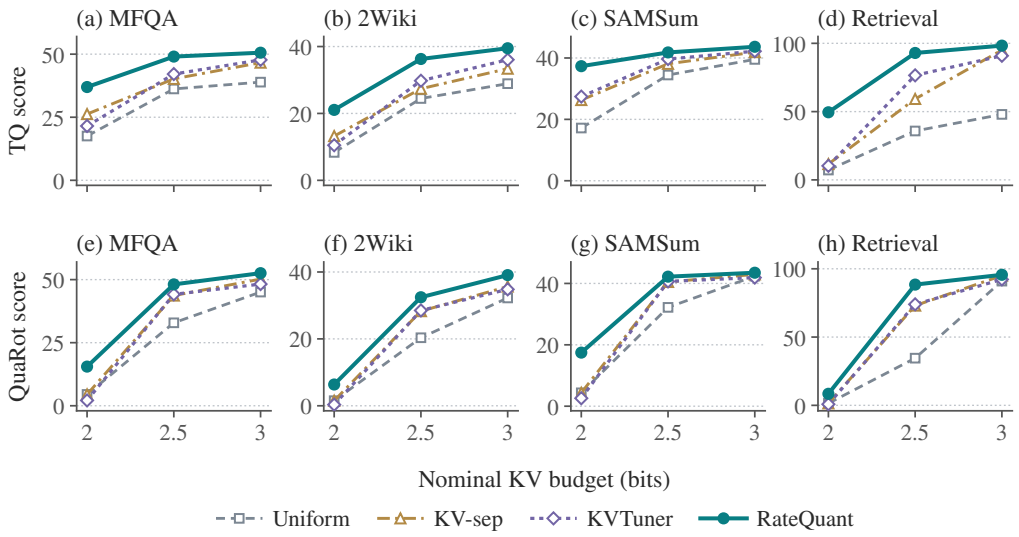


Figure 1: **Frozen allocations improve long-context task quality.** Mean scores across three models for four LongBench tasks, using the same WikiText-2-calibrated bit tables across tasks. Rows show the base quantizer; each task shares a vertical scale across rows. Points are measured budgets, joined to guide comparison. Table 2 also reports code completion.

error. Conditional measurements retain this information rather than assigning a fixed importance to a component independently of the current bit table. An allocation criterion can use these signed responses directly.

We propose **RATEQUANT**, which refines a mixed-precision KV allocation using *conditional attention-distortion curves*. Each round measures adjacent-bit alternatives while fixing the inputs, residual normalization, and quantizer state at the current allocation. An exact dynamic program combines the measured costs into a joint proposal with balanced bit transfers. The proposal becomes the next exploration anchor, where the curves are remeasured. Task loss on a separate calibration set retains the best complete allocation visited. This separates inexpensive local proposal construction from task-level selection, without requiring a backward pass or a full-network loss evaluation for each component.

Our analysis follows this separation. A hybrid-network argument relates normalized attention distortion to task loss under bounded suffix sensitivity. A decomposition distinguishes measured single-component changes from joint interactions and changes in downstream inputs. Dynamic programming exactly optimizes the local additive objective in a bounded adjacent-bit neighborhood. Remeasurement then updates the costs as the quantized computation changes, while the final selection uses the complete model.

Across three models and two base quantizers, RATEQUANT improves WikiText-2 perplexity over uniform allocation in all 18 tested settings. At 2.5 bits, the reduction relative to the strongest tested baseline ranges from 3.2% to 10.3%. The allocations are then frozen and transferred to five LongBench tasks (Figure 1), where RATEQUANT achieves the highest three-model mean in every task, backend, and budget combination. Additional measurements examine the structure of the selected bit tables and the measured attention distortion of joint proposals. The resulting static table requires no allocation search during inference.

Our contributions are as follows:

- **Conditional attention-distortion measurements.** We connect normalized attention error to task loss and measure how finite precision changes interact with the current residual.
- **Joint allocation with iterative remeasurement.** We combine exact budget-preserving local optimization with full-network selection on separate calibration data.
- **Quality and transfer across quantizers.** Language modeling, frozen-allocation LongBench evaluation, and allocation diagnostics support attention-aware precision assignment.

2 ATTENTION DISTORTION UNDER A BIT BUDGET

We first define attention reconstruction at the inputs produced by a quantized prefix and relate this objective to task loss.

2.1 A COMMON-INPUT RECONSTRUCTION OBJECTIVE

For L layers and H KV heads per layer, let $n = 2LH$ index the K and V components separately. An allocation is $\mathbf{b} \in \mathbb{Z}^n$. For equal-size components, the feasible set is

$$\mathcal{B} = \{\mathbf{b} : \; b_{\min} \le b_i \le b_{\max}, \; \sum_{i=1}^{n} b_i = B\}, \qquad B = n\bar{b} \in \mathbb{Z}. \tag{1}$$

We require $nb_{\min} \le B \le nb_{\max}$ and count nominal KV data bits, excluding quantizer metadata. A weighted constraint extends the formulation to unequal-size components (Section D).

Let x denote a sequence and its next-token targets, and let $h_\ell^{\mathbf{b}}(x)$ be the residual-stream input to layer ℓ after executing the preceding layers with allocation $\mathbf{b}$. Write $A_\ell(h)$ for the unquantized attention output, including the output projection, and $\widehat{A}_\ell(h; \mathbf{b}_\ell)$ for its KV-quantized counterpart at the *same* input. All other model operations are held fixed. Define

$$e_\ell(\mathbf{b}, x) = \widehat{A}_\ell(h_\ell^{\mathbf{b}}(x); \mathbf{b}_\ell) - A_\ell(h_\ell^{\mathbf{b}}(x)). \tag{2}$$

Both attention outputs use the input produced by the same quantized prefix, isolating the error introduced at layer ℓ.

For an input $h \in \mathbb{R}^{S\times d}$, set $s_t(h)^2 = \max\{d^{-1}\sum_j h_{tj}^2, \epsilon\}$ with fixed $\epsilon > 0$, and define the linear normalization operator

$$[\mathcal{N}_h z]_{tj} = \frac{z_{tj}}{\sqrt{Sd}\, s_t(h)}. \tag{3}$$

This operator normalizes by residual-stream scale and averages over tokens and coordinates. With $r_\ell(\mathbf{b}, x) = \mathcal{N}_{h_\ell^{\mathbf{b}}(x)} e_\ell(\mathbf{b}, x)$, define

$$D_\ell(\mathbf{b}) = \mathbb{E}_{x\sim\mathcal{D}}\|r_\ell(\mathbf{b}, x)\|_F^2, \qquad \min_{\mathbf{b}\in\mathcal{B}} \mathcal{R}(\mathbf{b}) := \sum_{\ell=1}^{L} D_\ell(\mathbf{b}). \tag{4}$$

The analysis uses a fixed distribution $\mathcal{D}$; implementation averages the objective and curves over the same finite calibration set $\mathcal{D}_c$. Components interact through attention and through the dependence of downstream inputs on upstream allocations.

Connection to task loss. Let $\mathcal{L}_x(\mathbf{b})$ and $\mathcal{L}_x(\mathrm{FP})$ be quantized and unquantized sequence-mean cross-entropy. Assume that, on the comparison domain, an attention-output perturbation at layer ℓ changes loss through the remaining unquantized computation by at most C_ℓ times its normalized magnitude. Appendix A.1 states this uniform suffix-sensitivity assumption precisely.

Proposition 1 (Loss–distortion bound)**.** *Under Assumption 3, for every feasible allocation in the comparison domain,*

$$\left|\mathbb{E}_x[\mathcal{L}_x(\mathbf{b}) - \mathcal{L}_x(\mathrm{FP})]\right| \le \sum_\ell C_\ell \sqrt{D_\ell(\mathbf{b})} \le \|\mathbf{C}\|_2 \sqrt{\mathcal{R}(\mathbf{b})}. \tag{5}$$

The proof telescopes between networks with successively longer quantized prefixes. Consecutive networks share the changed layer's input, so the sensitivity bound applies to the residual in Eq. (2). This connection motivates attention distortion for proposal construction; full-model calibration loss selects among complete allocations.

3 CONDITIONAL CURVES AND JOINT ALLOCATION

At an anchor $\mathbf{b}^0$, run the quantized model on the curve-calibration set and freeze each observed $h_\ell^0 = h_\ell^{\mathbf{b}^0}(x)$ for its local comparisons. For component i in layer ℓ, the measured choices are

$$\mathcal{S}_i(\mathbf{b}^0) = \{b_i^0 - 1, b_i^0, b_i^0 + 1\} \cap [b_{\min}, b_{\max}].$$

Let $\mathbf{b}_\ell^{0,i\to u}$ replace only component i by u, and set

$$r_\ell^0(x) = r_\ell(\mathbf{b}^0, x), \tag{6}$$

$$\delta_i(u,x) = \mathcal{N}_{h_\ell^0}\big[\widehat{A}_\ell(h_\ell^0; \mathbf{b}_\ell^{0,i\to u}) - \widehat{A}_\ell(h_\ell^0; \mathbf{b}_\ell^0)\big]. \tag{7}$$

The local distortion curve records the resulting error change:

$$c_i(u \mid \mathbf{b}^0) = \mathbb{E}_x\big[2\langle r_\ell^0, \delta_i(u)\rangle_F + \|\delta_i(u)\|_F^2\big]. \tag{8}$$

Equivalently, $c_i(u \mid \mathbf{b}^0) = \mathbb{E}_x[\|r_\ell^0 + \delta_i(u)\|_F^2 - \|r_\ell^0\|_F^2]$, with $c_i(b_i^0 \mid \mathbf{b}^0) = 0$.

A negative value indicates reduced reconstruction error; a positive value indicates increased error. The cross term measures how the update reinforces or cancels the current residual, while the squared term measures the update's own magnitude. Figure 2 illustrates one sample with $\|r\|_F = \|\delta\|_F = 1$. For an angle θ between them, the cost is $1 + 2\cos\theta$ and changes sign at $120°$. The same update magnitude can therefore increase or decrease distortion. The empirical curves in Eq. (8) can be signed and nonmonotone, so the allocator uses their values directly without imposing a shared decay law.

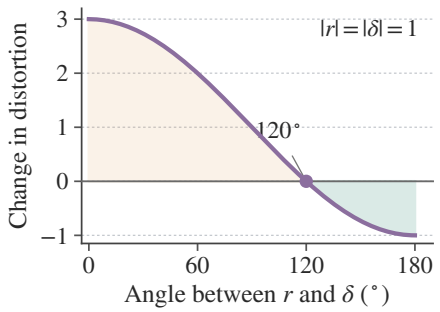


Figure 2: **Direction changes allocation value.** Analytic unit-norm case of Eq. (8).

During a local comparison, the input, quantizer state, and all other components remain fixed. Each alternative recomputes the affected attention group and its contribution through the output projection. For grouped-query attention, this includes all query heads sharing the modified KV head. Curve construction therefore uses local attention computations without a backward pass or a full-network loss evaluation for each component.

From local curves to joint updates. For a simultaneous proposal, let $\delta_i = \delta_i(b_i, x)$, and let η_ℓ be the difference between the joint normalized attention change at h_ℓ^0 and $\sum_{i\in\ell}\delta_i$. Define $\mathcal{R}_0(\mathbf{b} \mid \mathbf{b}^0)$ by evaluating (4) with all inputs and normalization scales frozen at the anchor. Then

$$\begin{aligned}\mathcal{R}(\mathbf{b}) - \mathcal{R}(\mathbf{b}^0) = &\sum_i c_i(b_i \mid \mathbf{b}^0) \\ &+ E_{\text{joint}}(\mathbf{b}; \mathbf{b}^0) + E_{\text{input}}(\mathbf{b}; \mathbf{b}^0),\end{aligned} \tag{9}$$

Here E_{joint} collects cross terms between component changes and nonadditive K/V interactions within a head. The term E_{input} accounts for changes in downstream inputs and their normalization. Appendix A.3 expands both terms. This decomposition motivates a bounded proposal followed by remeasurement at the new anchor: the proposal controls which components change, and remeasurement updates the inputs and residuals used to score the next one.

3.1 Joint Bit Allocation

A transfer decreases one component by one bit and increases another by one bit. With transfer cap M, the local feasible set is

$$\mathcal{U}_M(\mathbf{b}^0) = \left\{ \mathbf{b} \in \mathcal{B} : \begin{array}{l} b_i - b_i^0 \in \{-1, 0, 1\} \quad \text{for every } i, \\ \#\{i : b_i < b_i^0\} = \#\{i : b_i > b_i^0\} \le M \end{array} \right\}. \tag{10}$$

We solve

$$\widehat{\mathbf{b}} \in \operatorname*{arg\,min}_{\mathbf{b}\in\mathcal{U}_M(\mathbf{b}^0)} \sum_i c_i(b_i \mid \mathbf{b}^0). \tag{11}$$

A proposal can change up to $2M$ components through M donors and M recipients. The unchanged allocation is feasible, and the optimizer may use fewer than M transfers.

Let $F_i(d, u)$ be the minimum curve sum for the first i components with exactly d decrements and u increments. Starting from $F_0(0,0) = 0$, each state considers the three allowed choices for component i. After all components, we choose the smallest $F_n(k,k)$ over $0 \le k \le M$ and backtrack. The diagonal enforces the exact bit budget, while optimizing over k chooses the number of transfers.

Algorithm 1 RATEQUANT refinement under a fixed bit budget

Require: $\mathbf{b}^0 \in \mathcal{B}$, sets $\mathcal{D}_c, \mathcal{D}_s$, caps M, T; fixed settings
$\mathbf{a} \leftarrow \mathbf{b}^0$; $\mathbf{b}^{\text{best}} \leftarrow \mathbf{b}^0$; $\mathcal{V} \leftarrow \varnothing$
$\ell_{\text{best}} \leftarrow \text{CachedLoss}(\mathbf{b}^0)$
for $t = 1, \ldots, T$ **do**
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{\mathbf{a}\}$; measure $c_i(\cdot \mid \mathbf{a})$ on $\mathcal{D}_c$
  $\widehat{\mathbf{b}} \leftarrow \text{DP}(c, \mathbf{a}, M)$ using the local objective in Eq. (11)
  $\widehat{\ell} \leftarrow \text{CachedLoss}(\widehat{\mathbf{b}})$ on the separate selection set $\mathcal{D}_s$
  if $\widehat{\ell} < \ell_{\text{best}}$ **then**
    $\mathbf{b}^{\text{best}} \leftarrow \widehat{\mathbf{b}}$; $\ell_{\text{best}} \leftarrow \widehat{\ell}$
  end if
  if $\widehat{\mathbf{b}} \in \mathcal{V}$ **then break**
  $\mathbf{a} \leftarrow \widehat{\mathbf{b}}$; remeasure at this anchor in the next round
end for
return $\mathbf{b}^{\text{best}}$, frozen for test evaluation

Proposition 2 (Exact local allocation)**.** *For finite measured costs, this dynamic program minimizes* (11) *over* $\mathcal{U}_M(\mathbf{b}^0)$ *in exact arithmetic. It uses* $O(n(M+1)^2)$ *time and requires neither convexity nor monotonicity of the curves.*

The state records all constraints relevant to the remaining components; the additive objective then gives optimal substructure. Appendix A.2 gives the recurrence and proof. Signed costs cause no difficulty, and the unchanged allocation provides a feasible zero-cost proposal. The exactness concerns the measured local objective; the coupled objective in (4) is related through (9).

If $|E_{\text{joint}} + E_{\text{input}}| \le \varepsilon$ uniformly over $\mathcal{U}_M(\mathbf{b}^0)$, the exact surrogate minimizer is within 2ε of the minimum of $\mathcal{R}$ in that neighborhood, with $\mathcal{R}$ and the curves evaluated on the same distribution (Proposition 4, Appendix A.3).

3.2 ITERATIVE REFINEMENT

We use disjoint curve and selection sets, $\mathcal{D}_c$ and $\mathcal{D}_s$. The curves are measured on $\mathcal{D}_c$, while the deterministic empirical loss $\widehat{\mathcal{L}}_s$ on $\mathcal{D}_s$ selects the final allocation. Model evaluation settings and quantizer randomness remain fixed. In Algorithm 1, CachedLoss returns this selection loss and evaluates each distinct allocation only once. Initialization $\mathbf{b}^0 \in \mathcal{B}$ and hyperparameters are chosen using calibration data.

Exploration and selection. Algorithm 1 maintains an exploration anchor $\mathbf{a}$ and the best evaluated allocation $\mathbf{b}^{\text{best}}$. The anchor advances to each proposal even when its selection loss is higher; the retained allocation changes only on a strict improvement. Curves are then remeasured at the new anchor. An unchanged proposal, return to an expanded anchor, or the cap of T proposals ends the search.

Every proposal preserves the bit budget, and the best recorded selection loss is nonincreasing. The returned allocation is frozen before final test evaluation.

Calibration computation. Each round measures at most $2n$ local alternatives. DP storage is $O((M+1)^2)$ for rolling scores and $O(n(M+1)^2)$ for backtracking. Initialization and proposals require at most $T+1$ selection evaluations; Appendix C.2 includes the separate pool-initialization and curve-construction costs.

4 EXPERIMENTS

We evaluate three questions: whether conditional allocation improves language modeling at a fixed nominal bit budget, whether the frozen bit tables transfer to long-context tasks, and how the resulting proposals and allocations behave.

Table 1: **Perplexity across quantizers and budgets.** WikiText-2 test PPL (↓); subcolumns give nominal average KV bits. Bold marks the lowest value within each configuration, including ties.

Allocation	Qwen3-4B			Qwen3-8B			Llama-3.1-8B		
	2	2.5	3	2	2.5	3	2	2.5	3
TQ									
Uniform	323.66	25.49	23.58	61.34	11.81	10.96	12.18	8.46	7.83
KV-sep	85.28	29.30	15.54	20.54	11.19	10.23	12.18	8.27	7.81
KVTuner	19.97	14.69	13.34	13.62	10.55	9.72	16.98	8.76	7.80
RATEQUANT	**17.10**	**13.84**	**12.97**	**11.90**	**10.04**	**9.70**	**9.50**	**7.99**	**7.63**
QuaRot									
Uniform	3111.04	21.96	15.31	383.61	13.42	10.16	31.86	9.24	7.88
KV-sep	3111.04	17.48	14.46	383.61	10.76	9.80	31.86	8.52	7.88
KVTuner	585.05	17.23	14.47	530.10	10.73	10.11	280.78	8.53	8.09
RATEQUANT	**32.39**	**15.45**	**14.12**	**21.18**	**10.34**	**9.63**	**14.11**	**8.25**	**7.64**

4.1 EXPERIMENTAL SETUP

Models and quantizers. We use Qwen3-4B, Qwen3-8B, and Llama-3.1-8B-Instruct with TURBOQUANT (TQ) and QuaRot (Yang et al., 2025; Llama Team, 2024; Zandieh et al., 2026; Ashkboos et al., 2024). We adapt both KV quantizers to per-head mixed precision while keeping weights and other operations in FP16. The nominal average KV budgets are 2, 2.5, and 3 bits, with component precisions in $[1, 6]$. Quality is evaluated using eager KV quantization and dequantization. Appendix B gives execution settings and the distinction between nominal data bits and physical cache storage.

Calibration and baselines. All allocation choices use WikiText-2 training data (Merity et al., 2017). Disjoint sets of 8, 16, and 64 windows, each 2048 tokens long, provide pool initialization, local curves, and selection loss. *Uniform* uses equal precision at integer budgets; at 2.5 bits, it assigns 3 bits to the first half of each K/V pool in layer-major order and 2 bits to the rest. *KV-sep* chooses a global K/V split from pool-level calibration. Our *KVTuner* adaptation searches grouped-layer K/V pairs under the same nominal bit cap (Li et al., 2025). Its realized precision may fall below the cap (Appendix E.2), whereas RATEQUANT preserves the full budget. RATEQUANT starts from KV-sep and uses at most four proposals, each with at most 32 balanced transfers. Complete allocations are selected by cross-entropy on the separate 64-window set; test data are used only after this choice is frozen.

Evaluation. WikiText-2 test PPL uses the first 50 nonoverlapping windows of 2048 tokens. For LongBench (Bai et al., 2024), we reuse the frozen allocations with shared prompts, greedy generation, and task-specific scoring. The five tasks cover document QA, multi-document QA, dialogue summarization, passage retrieval, and code completion. For each model, task scores are averaged over three evaluation seeds (40, 41, and 42); the main table then reports the equally weighted mean across the three models. Appendix E.1 gives generation settings and every per-model score.

4.2 LANGUAGE MODELING ACROSS PRECISION BUDGETS

Table 1 compares uniform precision, calibrated global K/V splitting, grouped-layer search, and RATEQUANT allocation across layers and heads under shared nominal bit caps. RATEQUANT achieves lower PPL than all three baselines in each of the 18 model–quantizer–budget settings.

Fine-grained allocation is most valuable at tight budgets. At 2 bits under TQ, RATEQUANT reduces PPL from the strongest tested baseline's 19.97 to 17.10 on Qwen3-4B and from 13.62 to 11.90 on Qwen3-8B. Under QuaRot, the corresponding comparisons are 585.05 to 32.39 and 383.61 to 21.18. A global K/V split alone leaves the QuaRot 2-bit results unchanged from uniform allocation. Assigning precision within the pools therefore provides a degree of freedom that is consequential at this budget.

At 2.5 bits, RATEQUANT improves on the strongest tested baseline by 3.2–10.3% across the six model–quantizer combinations. For example, Qwen3-4B under QuaRot improves from 17.23 to

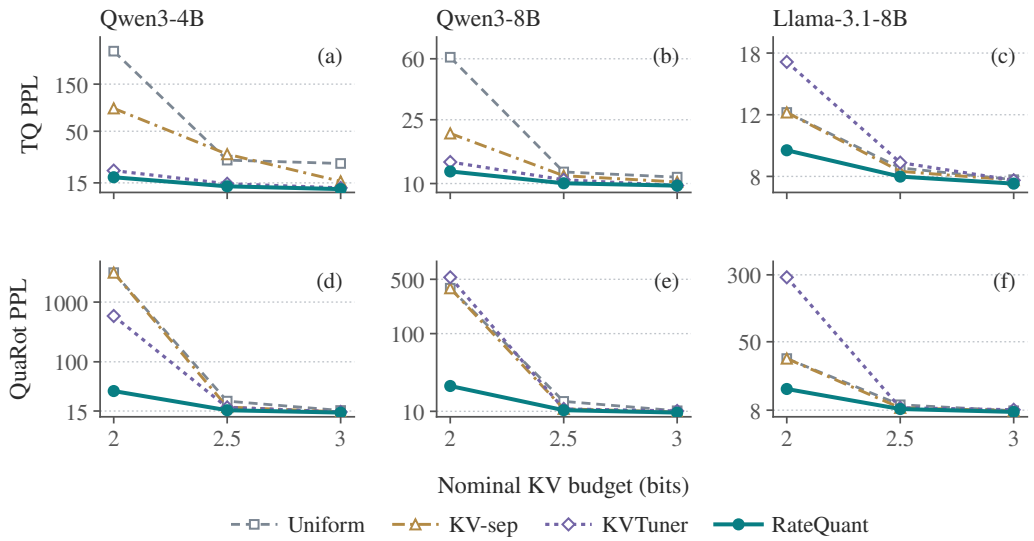


Figure 3: **Allocation gains vary with the precision budget.** WikiText-2 PPL for all models, quantizers, and tested budgets. Lower is better. Each panel has its own logarithmic vertical range; lines join measured budgets. Table 1 gives exact values.

Table 2: **Transfer of frozen allocations to LongBench.** Scores (↑), averaged equally over the three models for each task; bold marks the best mean within each backend and budget. MFQA is MultiFieldQA-en, 2Wiki is 2WikiMQA, and Retr. is PassageRetrieval-en.

Bits	Allocation	TQ					QuaRot				
		MFQA	2Wiki	SAMSum	Retr.	LCC	MFQA	2Wiki	SAMSum	Retr.	LCC
2	Uniform	17.55	8.35	17.18	7.21	33.56	4.49	1.52	4.36	1.45	22.10
	KV-sep	26.25	13.23	26.30	11.59	37.73	4.49	1.52	4.36	1.45	22.10
	KVTuner	21.53	10.49	27.42	10.27	42.65	2.11	0.33	2.59	0.82	22.53
	RateQuant	**36.97**	**21.06**	**37.33**	**49.51**	**46.92**	**15.52**	**6.36**	**17.46**	**8.46**	**28.98**
2.5	Uniform	36.22	24.47	34.46	35.81	50.40	32.88	20.34	32.20	34.54	45.99
	KV-sep	40.21	27.39	38.10	59.27	55.06	43.68	28.29	40.48	73.04	57.80
	KVTuner	42.11	29.77	39.65	76.56	52.19	44.16	28.50	40.65	73.97	57.81
	RateQuant	**49.04**	**36.28**	**41.81**	**92.95**	**58.97**	**48.11**	**32.45**	**42.21**	**88.42**	**58.52**
3	Uniform	38.91	28.90	39.54	47.96	57.56	45.07	32.24	42.38	90.98	61.13
	KV-sep	46.64	33.38	41.91	95.08	62.18	50.21	35.65	42.95	95.00	61.81
	KVTuner	47.73	36.06	42.21	90.89	58.98	48.23	34.81	41.90	92.22	58.33
	RateQuant	**50.58**	**39.49**	**43.65**	**98.31**	**64.18**	**52.56**	**39.07**	**43.49**	**95.60**	**62.53**

15.45, while Llama under TQ improves from 8.27 to 7.99. Higher budgets bring the methods closer together as quantization error decreases. Figure 3 shows this trend across all three measured budgets.

4.3 Transfer to Long-Context Tasks

The LongBench evaluation tests whether allocations calibrated on short language-modeling windows also help longer-context generation. No LongBench examples are used to measure curves or choose the bit table. The same allocation is reused across all five tasks, allowing the task results to evaluate transfer from the WikiText-2 calibration objective.

Table 2 reports the mean across models separately for each task. RateQuant has the highest mean in all 30 task–quantizer–budget combinations. The improvements span QA, summarization,

Table 3: **Measured quality of joint proposals.** Distortion reduction from the anchor and gap to the best of 16 evaluated candidates (%), at 2.5 bits. Each model–quantizer pair uses its first two refinement anchors.

Quantizer	Round	Qwen3-4B		Qwen3-8B		Llama-3.1-8B	
		Reduction	Gap	Reduction	Gap	Reduction	Gap
TQ	1	57.55	0.029	53.35	0.003	19.85	0.109
	2	65.61	0.102	51.79	0.000	9.70	0.032
QuaRot	1	67.53	0.081	65.97	0.013	27.66	0.190
	2	27.44	0.484	26.38	0.235	10.72	0.245

retrieval, and code completion, rather than being confined to one scoring rule. The per-model tables in Appendix E.1 expose the individual comparisons behind these means.

The gains persist beyond perplexity. At 2 bits under TQ, MultiFieldQA-en increases from the strongest baseline's 26.25 to 36.97, and passage retrieval from 11.59 to 49.51. At 2.5 bits, retrieval rises from 76.56 to 92.95 under TQ and from 73.97 to 88.42 under QuaRot. This benefit remains visible at 3 bits: TQ's 2WikiMQA mean increases from 36.06 to 39.49 and its LCC code score from 62.18 to 64.18. These comparisons use each task's strongest tested baseline within the same backend and budget.

4.4 Structure of the Learned Allocations

For Qwen3-4B at 2.5 bits, TQ and QuaRot start from K3/V2 but produce different layer–head assignments. Under TQ, all eight first-layer key heads receive 6 bits and the value heads receive 4–5 bits. Under QuaRot, the first-layer keys receive 4 bits and values receive 3 bits. The first layer contains only 2.8% of the K/V components but receives 67.2% of the positive bit increments under TQ and all of them under QuaRot.

The corresponding reductions are also backend-dependent: 64 key components move from 3 to 2 bits under TQ, compared with 16 under QuaRot. A common starting K/V split and a common total budget therefore need not imply similar final head assignments. The measured responses determine both which components donate precision and where the released bits go.

Across the full evaluation, all 18 selected allocations differ from their initialization. The selected tables concentrate extra precision early while reallocating bits elsewhere to preserve the exact overall budget. This pattern concerns the observed assignments; the objective itself measures conditional attention responses at every layer.

4.5 Measured Quality of Joint Proposals

We assess joint proposals at the first two refinement anchors for each model and backend at 2.5 bits, giving 12 comparisons. The diagnostic uses precision range $[1, 6]$ and transfer cap $M = 32$. At each anchor, 16 candidates comprise the DP proposal, the unchanged anchor, eight k-best alternatives of the same additive objective, four random balanced transfers, and two structured transfers. All candidates are measured on 16 calibration windows under their own quantized prefixes, including the resulting input changes. Appendix A.4 details the protocol.

Table 3 reports a 9.70–67.53% reduction in measured attention distortion from the anchor. The DP proposal's relative gap to the best measured candidate has a median of 0.092% and remains below 0.49%. These results show that the local objective constructs proposals that substantially reduce coupled attention distortion and remain close to the best evaluated alternatives.

5 Related Work

KV representations and allocation. KIVI, KVQuant, QuaRot, and TurboQuant improve low-precision representations through K/V-specific granularity, outlier handling, or rotations (Liu et al., 2024; Hooper et al., 2024; Ashkboos et al., 2024; Zandieh et al., 2026). Precision allocation

complements these designs. KVmix ranks layer sensitivity using gradients (Li et al., 2026); KVTuner uses relative attention-output error to prune precision pairs and group layers, followed by configuration search (Li et al., 2025). Channel-level approaches offer another granularity (Liao & Wen, 2026; Xia et al., 2026; Zhang et al., 2026b). RATEQUANT measures finite adjacent-bit actions at KV-head granularity, retaining their interaction with the existing projected attention residual.

Rate–distortion allocation. Rate–distortion theory formalizes reconstruction under a storage constraint (Cover & Thomas, 2006). Radio optimizes weight quantization through rate–distortion (Young, 2025); BAQ uses Hessian-weighted distortion under a bit budget (Zhang et al., 2025). HAWQ-V2 uses Hessian-trace sensitivity (Dong et al., 2020), whereas HAWQ-V3 selects hardware-aware precision through integer programming (Yao et al., 2021). Recent work also applies rate–distortion allocation to the KV cache. RDKV combines attention-derived importance with calibrated distortion tables, including eviction as a zero-bit choice (Zhang et al., 2026a). AATC models expected attention distortion under quantization-noise assumptions and allocates bits to transformed channels (Laus et al., 2026). KVTC uses dynamic programming to assign precision to decorrelated cache coordinates for storage and transfer (Staniszewski & Łańcucki, 2026); KV-COBRA jointly assigns rank and precision within heads and redistributes the budget across heads (Ha et al., 2026). RATEQUANT uses measured finite responses of the base quantizer and refreshes them after each joint allocation change. Its DP optimizes these signed local costs, with full-network calibration loss selecting the final table.

Reconstruction and conditional calibration. Local output reconstruction guides post-training quantization in AdaRound and BRECQ (Nagel et al., 2020; Li et al., 2021). In KV compression, CAOTE and OBCache score token removal through changes in attention outputs (Goel et al., 2025; Gu et al., 2026), while HeadWiseKV calibrates per-head history lengths sequentially under the selected lower-layer prefix (Xie et al., 2026). These methods make attention quality or the current prefix relevant to allocation. RATEQUANT studies finite bit-width actions: it holds the current input fixed for each local comparison, retains the cross term with the existing projected residual, and then remeasures at the next anchor. This action-level measurement supports signed, nonmonotone responses across different quantizers and connects them to a joint integer-budget proposal.

6 DISCUSSION

The results support precision allocation as a design choice alongside the quantizer. Quantizer-specific bit tables learned from the same K3/V2 initialization show how allocation adapts to the quantizer. The LongBench results show that a static allocation selected on language-modeling windows can benefit longer-context tasks without task-specific recalibration.

The value of an extra bit depends on the allocation around it: a precision change can reinforce or cancel the existing attention residual, while upstream changes alter the inputs used to evaluate later layers. Remeasuring these finite responses lets each round adapt to the current quantized computation, and dynamic programming coordinates the changes under the shared bit budget.

Calibration and deployment. Refinement evaluates at most five complete allocations on the selection set; pool initialization, input collection, and local attention measurements contribute additional work. Inference then uses only the fixed bit table. The experiments assess quality with eager quantization and dequantization; realizing the nominal budget in a packed cache requires a storage layout and mixed-precision kernels. Appendices C.2 and D detail calibration costs and integration choices.

7 CONCLUSION

RATEQUANT allocates KV precision through finite attention responses, budget-preserving dynamic programming, and iterative remeasurement. The theory connects normalized attention reconstruction to task loss and separates exact local allocation from joint-update effects. Across three models and two quantizers, experiments demonstrate improved perplexity, effective joint proposals, and transfer to five LongBench tasks. RATEQUANT thus makes precision allocation an explicit optimization target alongside KV quantizer design, with attention responses guiding where limited precision is most valuable.

AI USE STATEMENT

AI-assisted tools were used for manuscript writing and language polishing. The authors reviewed and verified the final text and take responsibility for its content.

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.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.172.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006. doi: 10.1002/047174882X.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18518–18529, 2020.

Raghavv Goel, Junyoung Park, Mukul Gagrani, Dalton Jones, Matthew Morse, Harper Langston, Mingu Lee, and Chris Lott. CAOTE: KV Cache Selection for LLMs via Attention Output Error-Based Token Eviction. *arXiv preprint arXiv:2504.14051*, 2025.

Yuzhe Gu, Xiyu Liang, Jiaojiao Zhao, and Enmao Diao. OBCache: Optimal Brain KV Cache Pruning for Efficient Long-Context LLM Inference. In *International Conference on Machine Learning*, 2026.

Sihyeon Ha, Jaeho Lee, and Yo-Seb Jeon. KV-COBRA: KV Cache Compression via Co-Optimized Bit-Rank Allocation. *arXiv preprint arXiv:2609.24298*, 2026.

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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165.

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.

Fei Li, Song Liu, Weiguo Wu, Shiqiang Nie, and Jinyu Wang. KVmix: Gradient-based layer importance-aware mixed-precision quantization for KV cache. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(37):31563–31572, 2026. doi: 10.1609/aaai.v40i37.40422.

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, 2025.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the Limit of Post-Training Quantization by Block Reconstruction. In *International Conference on Learning Representations*, 2021.

Chengxi Liao and Zeyi Wen. Channel-aware mixed-precision quantization for efficient long-context inference. 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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344, 2024.

Llama Team. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or Down? Adaptive Rounding for Post-Training Quantization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 7197–7206. PMLR, 2020.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In *Proceedings of Machine Learning and Systems*, volume 5, 2023.

Konrad Staniszewski and Adrian Łańcucki. KV Cache Transform Coding for Compact Storage in LLM Inference. In *International Conference on Learning Representations*, 2026.

Haojun Xia, Xiaoxia Wu, Jisen Li, Tsai-chuan Wu, Junxiong Wang, Jue Wang, Chenxi Li, Aman Singhal, Alay Dilipbhai Shah, Alpay Ariyak, Donglin Zhuang, Zhongzhu Zhou, Ben Athiwaratkun, Zhen Zheng, and Shuaiwen Song. Kitty: Accurate and efficient 2-bit KV cache quantization with dynamic channel-wise precision boost. In *Proceedings of Machine Learning and Systems*, volume 8, 2026.

Renjie Xie, Juncheng Yang, Aoting Hu, Mingxi Zhang, Liyao Wu, Zheheng Hong, and Wei Xu. HeadWiseKV: Budgeted Per-Head Cache Residency for Hybrid Long-Context Language Models. *arXiv preprint arXiv:2609.02029*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, and Kurt Keutzer. HAWQ-V3: Dyadic Neural Network Quantization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 11875–11886. PMLR, 2021.

Sean I. Young. Radio: Rate–distortion optimization for large language model compression. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72819–72836, 2025.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online vector quantization with near-optimal distortion rate. In *International Conference on Learning Representations*, 2026.

Chao Zhang, Li Wang, Samson Lasaulce, and Merouane Debbah. BAQ: Efficient bit allocation quantization for large language models. *arXiv preprint arXiv:2506.05664*, 2025.

Junkai Zhang, Hang Guo, Luca Benini, and Yawei Li. RDKV: Rate-Distortion Bit Allocation for Joint Eviction and Quantization of the KV Cache. *arXiv preprint arXiv:2605.08317*, 2026a.

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, 2026b. doi: 10.18653/v1/2026.acl-long.326.

A Theoretical Analysis

We state and prove the loss–distortion bound, give the dynamic programming recurrence and its optimality proof, and expand the joint-update decomposition and its implication for local allocation accuracy.

A.1 Loss–Distortion Bound

For a fixed layer input h, define $G_{\ell,x,h}(z)$ as the loss obtained by injecting projected attention output z into that layer, retaining its residual and remaining operations, and using an unquantized suffix thereafter.

Assumption 3 (Uniform suffix sensitivity on the comparison domain)**.** *There exist finite constants $C_\ell \geq 0$, independent of x and $\mathbf{b}$ on the stated domain, such that, for $h = h_\ell^{\mathbf{b}}(x)$ and the two attention outputs in* (2)*,*

$$|G_{\ell,x,h}(z) - G_{\ell,x,h}(z')| \leq C_\ell \|\mathcal{N}_h(z - z')\|_F.$$

All expectations below are finite.

Proposition 1 follows from the hybrid-network argument below.

Proof. Fix a sequence x and an allocation $\mathbf{b}$. Let $f^{[\ell]}$ use quantized KV attention in layers $1, \ldots, \ell$ and unquantized attention thereafter, with all other operations unchanged. Then $f^{[0]} = f_{\mathrm{FP}}$ and $f^{[L]} = f_{\mathbf{b}}$. In the comparison $f^{[\ell]}$ versus $f^{[\ell-1]}$, both networks have the same quantized prefix through layer $\ell - 1$, and hence the same layer input $h_\ell^{\mathbf{b}}(x)$. Their attention outputs differ by precisely $e_\ell(\mathbf{b}, x)$. Their common suffix includes the same-layer residual addition and remaining sublayers as well as all subsequent unquantized layers. It is therefore the map $G_{\ell,x,h}$ in Assumption 3.

Writing $a_\ell(x)$ for the loss of $f^{[\ell]}$, telescoping gives

$$|\mathcal{L}_x(\mathbf{b}) - \mathcal{L}_x(\mathrm{FP})| = \left| \sum_{\ell=1}^{L} (a_\ell - a_{\ell-1}) \right| \leq \sum_{\ell=1}^{L} C_\ell \|r_\ell(\mathbf{b}, x)\|_F.$$

Taking expectations and applying Cauchy–Schwarz twice yields

$$\mathbb{E}_x|\mathcal{L}_x(\mathbf{b}) - \mathcal{L}_x(\mathrm{FP})| \leq \sum_\ell C_\ell \sqrt{\mathbb{E}_x \|r_\ell\|_F^2} \leq \left(\sum_\ell C_\ell^2 \right)^{1/2} \left(\sum_\ell D_\ell(\mathbf{b}) \right)^{1/2}.$$

Finally $|\mathbb{E}X| \leq \mathbb{E}|X|$ proves (5). □

Meaning of the sensitivity assumption. For fixed x, h, a sufficient condition is differentiability of $G_{\ell,x,h}$ along the segment from z' to z and a uniform bound on the gradient in normalized coordinates. If $z(q) = z' + \mathcal{N}_h^{-1} q$, it suffices that

$$\sup_{q \in [0, \mathcal{N}_h(z - z')]} \|\nabla_q G_{\ell,x,h}(z(q))\|_F \leq C_\ell.$$

Integration along the segment gives the required inequality. Since $\epsilon > 0$, $\mathcal{N}_h$ is invertible at each fixed h. The constants C_ℓ characterize suffix sensitivity uniformly over the comparison domain; allocation scores are obtained directly from the measured curves.

A.2 Dynamic Programming and Optimality

Let $F_i(d, u)$ be the minimum curve sum for the first i components using exactly d decrements and u increments. Initialize $F_0(0, 0) = 0$ and all other states to $+\infty$. For $0 \leq d, u \leq M$, use

$$F_i(d, u) = \min_{\substack{s \in \{-1,0,1\} \\ b_i^0 + s \in [b_{\min}, b_{\max}]}} \big\{ F_{i-1}(d - \mathbf{1}_{s=-1}, u - \mathbf{1}_{s=1}) + c_i(b_i^0 + s \mid \mathbf{b}^0) \big\}, \tag{12}$$

with out-of-range states assigned $+\infty$. Choose $k \in \arg\min_{0 \leq k \leq M} F_n(k, k)$ and backtrack.

Proposition 2 follows from the recurrence above.

Proof. Let a partial assignment specify the first i components. Its state records only the counts of decrements and increments, (d, u), since each component can change by at most one bit and all have equal storage weight. The base case $i = 0$ has only state $(0, 0)$ with cost zero. For the inductive step, removing component i from any feasible partial assignment leaves exactly the predecessor in (12), according to its last choice $s \in \{-1, 0, 1\}$. Conversely, every finite predecessor with a permitted choice produces a feasible partial assignment. Minimizing over the three choices therefore yields the optimal value for each state.

For a fixed state at a fixed i, all future options are identical. Retaining only its smallest additive cost cannot discard a better solution to the additive problem. Nonnegative counts never decrease during the forward recursion, so states exceeding M cannot lead to a feasible terminal solution. At $i = n$, the terminal diagonal $d = u$ enforces the budget, and minimizing over $0 \le d = u \le M$ enforces the cap. Recording one minimizing predecessor per state permits exact reconstruction. Ties may be resolved deterministically; the unchanged allocation remains available with score zero. The argument applies to arbitrary finite signed curve costs. □

The optimized domain $\mathcal{U}_M$ contains budget-balanced adjacent-bit updates. Changes of two or more bits in one component require multiple refinement rounds.

A.3 Joint-Update Analysis

At a fixed anchor, define the frozen-input objective explicitly by

$$\mathcal{R}_0(\mathbf{b} \mid \mathbf{b}^0) = \sum_{\ell} \mathbb{E}_x \left\| \mathcal{N}_{h_\ell^0} \left[\widehat{A}_\ell(h_\ell^0; \mathbf{b}_\ell) - A_\ell(h_\ell^0) \right] \right\|_F^2 .$$

For each layer, the joint normalized attention change is

$$\Delta_\ell = \mathcal{N}_{h_\ell^0} \left[\widehat{A}_\ell(h_\ell^0; \mathbf{b}_\ell) - \widehat{A}_\ell(h_\ell^0; \mathbf{b}_\ell^0) \right] = \sum_{i \in \ell} \delta_i + \eta_\ell .$$

The joint interaction term is

$$E_{\text{joint}} = \sum_{\ell} \mathbb{E}_x \left[2 \sum_{\substack{i<j \\ i,j \in \ell}} \langle \delta_i, \delta_j \rangle_F + 2 \left\langle r_\ell^0 + \sum_{i \in \ell} \delta_i, \eta_\ell \right\rangle_F + \|\eta_\ell\|_F^2 \right] . \tag{13}$$

Expand $\|r_\ell^0 + \Delta_\ell\|_F^2 - \|r_\ell^0\|_F^2$ and subtract the sum of the single-component changes. The terms left are exactly those in (13). Summing over layers and adding $E_{\text{input}} = \mathcal{R}(\mathbf{b}) - \mathcal{R}_0(\mathbf{b} \mid \mathbf{b}^0)$ gives (9), since $\mathcal{R}_0(\mathbf{b}^0 \mid \mathbf{b}^0) = \mathcal{R}(\mathbf{b}^0)$.

For distinct KV heads at fixed input and fixed group-local quantizer state, attention groups concatenate before a linear output projection, so their joint output changes add in real arithmetic. Even then, squared-error cross terms $\langle \delta_i, \delta_j \rangle_F$ need not vanish after projection. If both K and V of a head change, its attention weights and values change together; the resulting product need not equal the sum of the two single-component changes, and contributes to η_ℓ.

Bounding joint interactions. Let $a_i = (\mathbb{E}\|\delta_i\|_F^2)^{1/2}$, $v_\ell = (\mathbb{E}\|\eta_\ell\|_F^2)^{1/2}$, and $\rho_\ell = (\mathbb{E}\|r_\ell^0\|_F^2)^{1/2}$. Triangle and Cauchy–Schwarz inequalities give

$$|E_{\text{joint}}| \le \sum_{\ell} \left[2 \sum_{\substack{i<j \\ i,j \in \ell}} a_i a_j + 2 \left(\rho_\ell + \sum_{i \in \ell} a_i \right) v_\ell + v_\ell^2 \right] .$$

The bound expresses the joint contribution through the individual output changes, the anchor residuals, and the nonadditive output changes η_ℓ.

Proposition 4 (Conditional local approximation guarantee)**.** *Fix* $\mathbf{b}^0$ *and* $\mathcal{U}_M(\mathbf{b}^0)$. *Suppose the* total *surrogate remainder obeys*

$$|E_{\text{joint}}(\mathbf{b}; \mathbf{b}^0) + E_{\text{input}}(\mathbf{b}; \mathbf{b}^0)| \le \varepsilon \quad \textit{for every } \mathbf{b} \in \mathcal{U}_M(\mathbf{b}^0).$$

If $\widehat{\mathbf{b}}$ minimizes the additive surrogate and $\mathbf{b}^\star$ minimizes $\mathcal{R}$ on the same neighborhood, then

$$\mathcal{R}(\widehat{\mathbf{b}}) - \mathcal{R}(\mathbf{b}^\star) \le 2\varepsilon.$$

Proof. Write $P(\mathbf{b}) = \sum_i c_i(b_i \mid \mathbf{b}^0)$ and $E(\mathbf{b}) = E_{\text{joint}} + E_{\text{input}}$. Then

$$\mathcal{R}(\widehat{\mathbf{b}}) - \mathcal{R}(\mathbf{b}^\star) = P(\widehat{\mathbf{b}}) - P(\mathbf{b}^\star) + E(\widehat{\mathbf{b}}) - E(\mathbf{b}^\star) \le 2\varepsilon.$$

If the surrogate solver itself has additive suboptimality τ, the same argument gives $2\varepsilon + \tau$. □

The result uses the same distribution for $\mathcal{R}$ and the costs c_i, including the empirical distribution on $\mathcal{D}_c$. It quantifies local allocation accuracy in terms of the total remainder. For a population objective evaluated through empirical curves, the remainder also includes estimation error.

A.4 Empirical Quality of Joint Proposals

We evaluate the additive objective through the measured distortion of the joint allocation it selects. At each anchor, we compare the DP proposal with alternative budget-balanced allocations, measuring the improvement over the anchor and the gap to the best allocation in the evaluated candidate set.

We use the first two refinement anchors of Qwen3-4B, Qwen3-8B, and Llama-3.1-8B-Instruct under TQ and QuaRot at a 2.5-bit budget, giving 12 comparisons. Each anchor has 16 candidates fixed before distortion evaluation: the DP proposal, the unchanged anchor, eight further allocations obtained by a k-best extension of the same additive dynamic program, four random balanced proposals with 1, 4, 8, and M transfers, and two structured proposals with eight transfers each. The structured proposals couple K/V changes within heads or concentrate changes in the first quarter of layers. The comparisons use precision range $[1, 6]$ and transfer cap $M = 32$. All candidates are evaluated on the same 16 curve-calibration windows of 2048 tokens. Each candidate's distortion is measured under its own quantized prefix, including the resulting layer inputs and normalization scales.

Let $\mathcal{C}$ denote the candidate set at an anchor, let $\widehat{\mathbf{b}}$ be its DP proposal, and let $\mathcal{R}_c$ denote the measured attention distortion on these windows. We report the relative selection gap

$$g_{\mathcal{C}} = 100\,\frac{\mathcal{R}_c(\widehat{\mathbf{b}}) - \min_{\mathbf{b}\in\mathcal{C}} \mathcal{R}_c(\mathbf{b})}{\min_{\mathbf{b}\in\mathcal{C}} \mathcal{R}_c(\mathbf{b})}.$$

Both terms in the numerator are measured distortions. This quantity evaluates the quality of the selected allocation within $\mathcal{C}$.

The DP proposal reduces measured attention distortion in every comparison, by 9.70–67.53% from its anchor (Table 3). The median gap to the best measured candidate is 0.092%, and every gap is below 0.49%. Thus, even with close alternatives from the additive objective included in each comparison, the DP proposal achieves measured distortion close to the best evaluated candidate and substantially improves on the current allocation.

B Implementation and Evaluation Protocols

B.1 Calibration Data and PPL Evaluation

Allocation uses WikiText-2 training data with sampling seed 20260912 and quantizer seed 42. Pool calibration, local curves, and selection use disjoint sets of 8, 16, and 64 nonoverlapping 2048-token windows, respectively. Selection loss is the mean next-token cross-entropy on the fixed 64-window set. Test evaluation uses the first 50 nonoverlapping 2048-token WikiText-2 test windows, yielding 102350 prediction targets. Each configuration is frozen before test evaluation. The LongBench protocol and per-model scores are given in Section E.1.

B.2 Baseline Allocations

Uniform and KV-sep. Uniform assigns K2/V2 at 2 bits and K3/V3 at 3 bits. At 2.5 bits, the first half of the entries in each pool receive 3 bits and the remaining entries receive 2 bits in a deterministic

layer-major order. KV-sep measures whole-K-pool and whole-V-pool loss curves while keeping the other pool unquantized, then selects their budget-feasible combination. RATEQUANT initializes from this calibrated allocation.

KVTuner. Our KVTuner adaptation (Li et al., 2025) constructs candidate K/V pairs from measured distortion, applies Pareto filtering and layer grouping, and searches group-level precisions with NSGA-II. We use 100 trials, population size 10, search seed 0, 16 feature windows, and 64 selection windows of 2048 tokens. The search enforces the nominal average-bit budget as an upper bound; realized precisions are listed in Section E.2.

B.3 QUANTIZATION AND EXECUTION

TQ retains its fixed codebooks and rotation seed. QuaRot uses the official KV quantization and QK/OV rotation path adapted to each model and to per-head precision. The experiments keep weights and other operations in FP16 and evaluate KV quantization/dequantization with eager attention. All evaluations use batch size one, PyTorch 2.8.0+cu128, and Transformers 5.13.0 on NVIDIA H20, RTX PRO 6000, or A800 GPUs. All models have eight KV heads per layer; the Qwen models have 36 layers and Llama has 32.

B.4 REFINEMENT IMPLEMENTATION

Local measurements. At each exploration anchor, a forward pass collects layer inputs and baseline attention outputs on the fixed curve-calibration set. Candidate measurements reuse these inputs and normalization scales while recomputing the affected attention group. Quantizer randomness remains fixed, and the measured signed costs enter the dynamic program directly. New anchors require fresh inputs, residuals, and cost tables.

Joint proposals. We use $M = 32$ balanced transfers per proposal as an upper limit, with at most one bit of change per component. Dynamic programming selects the increments and decrements jointly. The DP table F_i is computed from F_{i-1}, so each component is assigned only once per round. Equal costs are resolved in a fixed iteration order, with one minimizing predecessor retained per state for backtracking. Larger changes in a component's precision require multiple rounds.

Exploration and stopping. The proposal becomes the next exploration anchor even when its selection loss is higher than the best recorded value. The retained allocation is updated only when selection loss improves; ties keep the earlier configuration. Search stops after at most $T = 4$ proposals, when the anchor is unchanged, or when an already expanded anchor is revisited. Stored selection losses are reused for repeated configurations. Thus, the retained loss is nonincreasing even when losses along the exploration path are not.

C CALIBRATION RECORDS AND COMPUTATIONAL COST

C.1 SELECTED ALLOCATION LOSS

We run up to four refinement rounds and select the allocation with the lowest selection-set cross-entropy among the initialization and evaluated proposals. Table 4 reports the loss of the selected allocation for each model, backend, and budget.

C.2 CALIBRATION COST

Calibration consists of initialization, local attention measurements, joint allocation, and selection-set evaluation. With $T = 4$, the refinement stage evaluates at most five complete allocations on the 64-window selection set, including initialization. The pool-calibration and local-measurement costs are accounted for separately below.

Curve construction. Let C_c be the cost of one forward execution on the curve-calibration set while collecting layer inputs and baseline attention outputs. Let $C_{\mathrm{loc},i}$ denote the cost of recomputing

Table 4: **Selection-set cross-entropy.** Loss of the retained allocation on each model's fixed 64-window training selection set; these are calibration losses, not test perplexities.

Model	Backend	2 bit	2.5 bit	3 bit
Qwen3-4B	TQ	2.925306	2.721326	2.703420
Qwen3-4B	QuaRot	3.566596	2.844372	2.760377
Qwen3-8B	TQ	2.557956	2.387134	2.350293
Qwen3-8B	QuaRot	3.116099	2.410840	2.338996
Llama-3.1-8B	TQ	2.307618	2.133658	2.087062
Llama-3.1-8B	QuaRot	2.679882	2.169439	2.090294

the affected attention group and accumulating its projected distortion change for one alternative of component i over that set. A round uses at most two alternatives per component, giving cost $C_c + O(\sum_i C_{\mathrm{loc},i})$. Each alternative is evaluated locally; its cost depends on sequence length, attention grouping, and implementation.

Allocation and selection. The DP costs $O(n(M+1)^2)$ time per round. Rolling score storage is $O((M+1)^2)$; predecessor storage is $O(n(M+1)^2)$. If C_s is the cost of one selection-set forward loss evaluation, at most T proposed allocations and one initial allocation require such evaluation. The total calibration cost is bounded by

$$C_{\mathrm{init}} + (T+1)C_s + O\left(T\left[C_c + \sum_i C_{\mathrm{loc},i} + n(M+1)^2\right]\right).$$

Here C_{init} is the cost of selecting the starting allocation. Caching and early termination reduce the realized cost.

D Scope and Extensions

Component size and precision schedule. The current budget and dynamic program assume equal-size K/V components with one static precision per component. Supporting unequal-size components requires storage weights in both the budget constraint and the allocation state. Token-dependent precision would additionally require a policy for varying assignments during inference.

Allocation search. The joint-update decomposition in Section A.3 identifies interaction terms that could be incorporated into richer proposal models. Evaluating multiple proposals at an anchor offers another way to broaden exploration. Both extensions would change the calibration cost; the present method uses one additive proposal per expanded anchor.

Inference integration. The selected table contains n fixed bit-width entries, so inference requires neither allocation search nor curve remeasurement. A packed mixed-precision cache would map these assignments to a physical cache layout. Actual memory use and throughput depend on quantizer metadata and the kernels used to execute that layout; the current experiments evaluate quality with eager KV quantization and dequantization.

E Additional Experimental Results

E.1 LongBench Protocol and Per-model Results

Task scores are computed with the shared evaluation pipeline: token F1 for the two QA tasks, first-line ROUGE-L for SAMSum, the LongBench retrieval scorer, and the LongBench code-similarity scorer for LCC. We retain the full input context. SAMSum uses the established first-line stopping rule; the generation caps are 128 tokens for the first four tasks and 64 for LCC. The five tasks contain 150, 200, 200, 200, and 500 examples. Each table below reports a single model, with each score averaged over three evaluation seeds (40, 41, and 42), and no averaging across tasks.

Table 5: **LongBench scores for Qwen3-4B.** Scores (↑) retain each task's metric; no averaging over tasks. Bold marks the highest reported score within each backend and budget.

Bits	Allocation	TQ					QuaRot				
		MFQA	2Wiki	SAMSum	Retr.	LCC	MFQA	2Wiki	SAMSum	Retr.	LCC
2	Uniform	1.73	0.89	7.86	1.83	22.53	0.65	0.20	0.70	1.23	9.69
	KV-sep	9.65	2.83	10.98	1.95	36.65	0.65	0.20	0.70	1.23	9.69
	KVTuner	19.90	10.94	24.36	7.81	41.37	2.42	0.46	4.62	1.20	24.00
	RATEQUANT	**29.50**	**19.17**	**33.10**	**23.94**	**43.31**	**9.59**	**1.88**	**8.91**	**2.10**	**29.47**
2.5	Uniform	17.74	7.27	23.82	5.19	43.69	16.67	8.71	22.81	13.46	37.82
	KV-sep	22.24	9.00	30.05	8.43	49.05	31.05	16.60	36.68	43.38	54.45
	KVTuner	34.42	20.61	36.37	50.37	47.49	32.37	17.10	37.24	45.63	54.47
	RATEQUANT	**45.54**	**33.85**	**39.56**	**85.29**	**55.02**	**42.07**	**27.70**	**40.96**	**80.75**	**54.59**
3	Uniform	21.05	11.01	32.55	13.26	51.38	34.48	22.56	40.52	77.71	58.84
	KV-sep	39.99	21.35	38.61	88.41	59.66	48.67	33.56	42.74	88.99	60.48
	KVTuner	41.54	28.92	39.85	76.42	50.29	44.47	28.18	40.35	86.97	54.85
	RATEQUANT	**46.98**	**32.24**	**42.64**	**96.50**	**60.91**	**49.59**	**34.06**	**43.19**	**89.90**	**61.08**

Table 6: **LongBench scores for Qwen3-8B.** Scores (↑) retain each task's metric; no averaging over tasks. Bold marks the highest reported score within each backend and budget.

Bits	Allocation	TQ					QuaRot				
		MFQA	2Wiki	SAMSum	Retr.	LCC	MFQA	2Wiki	SAMSum	Retr.	LCC
2	Uniform	8.62	1.85	6.68	0.29	38.50	1.02	0.15	1.88	0.09	30.15
	KV-sep	26.81	14.54	30.92	13.32	36.91	1.02	0.15	1.88	0.09	30.15
	KVTuner	23.27	10.00	33.79	19.41	42.76	2.74	0.36	1.95	0.29	19.45
	RATEQUANT	**33.81**	**18.85**	**39.47**	**54.75**	**43.35**	**13.53**	**5.04**	**22.63**	**4.80**	**30.79**
2.5	Uniform	39.51	27.63	37.55	15.75	51.53	33.54	18.54	33.61	31.97	46.47
	KV-sep	45.51	31.74	42.75	73.84	58.13	48.78	33.00	42.98	85.24	61.41
	KVTuner	41.73	29.01	42.04	85.33	53.41	48.93	33.19	42.99	85.77	61.54
	RATEQUANT	**47.61**	**32.65**	**43.79**	**96.58**	**62.02**	**49.79**	**33.70**	**43.14**	**89.83**	**62.17**
3	Uniform	43.97	33.58	42.12	35.12	61.50	48.42	36.40	43.45	97.23	65.56
	KV-sep	48.24	37.72	43.55	98.83	63.45	49.64	35.65	42.95	98.01	65.97
	KVTuner	47.16	34.32	43.81	98.25	64.89	48.10	35.99	42.10	94.83	60.80
	RATEQUANT	**49.63**	**38.47**	**44.26**	**99.43**	**67.78**	**52.48**	**37.63**	**43.83**	**98.39**	**66.24**

Table 7: **LongBench scores for Llama-3.1-8B-Instruct.** Scores (↑) retain each task's metric; no averaging over tasks. Bold marks the highest reported score within each backend and budget.

Bits	Allocation	TQ					QuaRot				
		MFQA	2Wiki	SAMSum	Retr.	LCC	MFQA	2Wiki	SAMSum	Retr.	LCC
2	Uniform	42.28	22.31	37.01	19.50	39.64	11.80	4.21	10.51	3.03	26.46
	KV-sep	42.28	22.31	37.01	19.50	39.64	11.80	4.21	10.51	3.03	26.46
	KVTuner	21.43	10.54	24.10	3.60	43.81	1.17	0.16	1.21	0.98	24.14
	RATEQUANT	**47.60**	**25.16**	**39.42**	**69.83**	**54.09**	**23.45**	**12.15**	**20.85**	**18.47**	**26.69**
2.5	Uniform	51.41	38.52	42.01	86.50	55.98	48.42	33.79	40.17	58.20	53.66
	KV-sep	52.87	41.42	41.50	95.53	58.01	51.21	35.26	41.78	90.50	57.55
	KVTuner	50.19	39.71	40.53	94.00	55.66	51.19	35.21	41.72	90.50	57.43
	RATEQUANT	**53.98**	**42.34**	**42.09**	**96.97**	**59.89**	**52.47**	**35.97**	**42.53**	**94.67**	**58.79**
3	Uniform	51.73	42.12	43.95	95.50	59.79	52.32	37.76	43.15	98.00	58.98
	KV-sep	51.70	41.06	43.59	98.00	63.44	52.32	37.76	43.15	98.00	58.98
	KVTuner	54.49	44.94	42.96	98.00	61.75	52.12	40.26	43.24	94.85	59.35
	RATEQUANT	**55.12**	**47.76**	**44.04**	**99.00**	**63.86**	**55.60**	**45.53**	**43.46**	**98.50**	**60.26**

E.2 REALIZED KVTUNER PRECISION

Table 8: **Realized KVTuner precision.** Average KV data bits under each nominal cap. These values describe the grouped-layer baseline, whose constraint permits unused budget.

Model	Backend	2 bit	2.5 bit	3 bit
Qwen3-4B	TQ	1.9028	2.3889	2.9722
Qwen3-4B	QuaRot	1.9444	2.5000	2.9861
Qwen3-8B	TQ	1.9861	2.4306	3.0000
Qwen3-8B	QuaRot	1.9583	2.5000	2.9861
Llama-3.1-8B	TQ	2.0000	2.5000	3.0000
Llama-3.1-8B	QuaRot	1.9688	2.5000	3.0000