KV-COBRA: KV Cache Compression via Co-Optimized Bit-Rank Allocation

Sihyeon Ha **Jaeho Lee** **Yo-Seb Jeon***
Pohang University of Science and Technology (POSTECH)
{sihyeon.ha, jaeho.lee, yoseb.jeon}@postech.ac.kr

Abstract

What limits KV-cache compression at extreme bit-rates? We argue that it is not the choice of compression scheme, but how its budget is allocated across attention heads. Existing methods apply rank and bit-width uniformly, ignoring that each head has a different optimal mix of rank truncation and quantization. We show that *co-optimizing rank and bit-width per head*—using only standard low-rank projection and scalar quantization—dominates uniform allocation, with the largest gains at low bit-rates. Our method, **KV-COBRA** (*Co-Optimized Bit-Rank Allocation*), formalizes this as a resource-allocation problem: it balances rank-truncation loss against quantization loss within each head, then redistributes budget across heads to minimize total distortion. A fused Hadamard rotation equalizes per-channel variance, and reordering the SVD basis by attention-KL importance makes the solver query-aware. The same allocator extends to joint $K+V$ compression. On perplexity, zero-shot, and long-context benchmarks from 0.5 to 4 bits per dimension (bpd), KV-COBRA shows the smallest accuracy degradation among evaluated methods at low bpd, with no per-token overhead.

1 Introduction

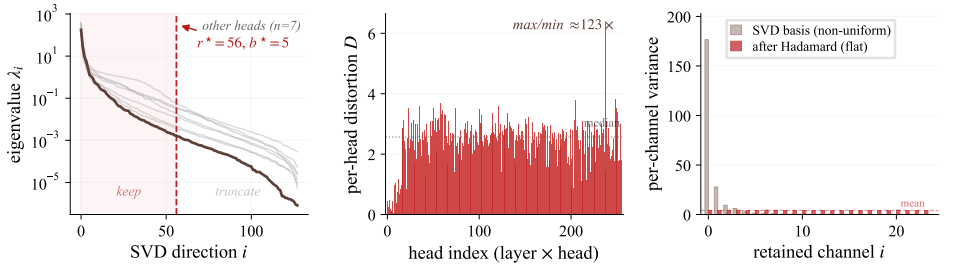
Generating each token with a large language model (LLM) requires loading the entire key-value (KV) cache. Under long-context scenarios, the size of these caches exceeds that of the model weights, rendering the decoding to be *memory-bandwidth-bound*: Loading the KV cache becomes the bottleneck, not computing on it [1, 2]. To alleviate this issue, two strategies to compress the KV cache have been proposed: *Eviction* discards KV for unimportant or redundant tokens [3–5]. *Per-token compression*, our focus, keeps every token but stores each at lower *bits per dimension* (bpd), via quantization [6–8], low-rank projection [9–11], or both. The strongest per-token compressors combine these, applying top-r SVD truncation followed by b-bit quantization on the retained coordinates.

In these per-token compression methods, the rank r and quantization bit-width b are typically set as global hyperparameters, applied uniformly across *all attention heads*. Yet, heads differ dramatically in how their cache spectrum concentrates; some place most of their energy in a handful of directions, while others spread it widely (Figure 1). Thus, applying a single r is wasteful for concentrated heads and inadequate for diffuse ones. Likewise, using a uniform bit-width b across heads may be suboptimal. Such homogeneous allocation may limit accuracy, especially in low-bpd regimes.

This raises two questions. First, *how should the total rate budget be allocated across heads?* Some heads may need higher precision, while others tolerate aggressive compression. Second, *how should each head's allotted rate split between r and b?* The two questions are coupled, since the optimal r for a head depends on its b and vice versa, so optimizing one while holding the other fixed cannot reach the joint optimum.

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) C1: joint $(r^\star, b^\star)$ choice **(b)** Heads differ: D under uniform 2 bpd **(c)** Hadamard: flattens per-channel variance

Figure 1: **(a) Spectrum sets $(r^\star, b^\star)$:** one head (dark) against the layer's 7 others (grey). **(b) Heads differ dramatically:** per-head D at uniform 2 bpd spans $100\times$ across 256 KV heads. **(c) Hadamard flattens variance:** per-channel, retained subspace, before (beige) and after (red).

We view this as a classical rate-distortion allocation problem [12, 13]. Each head has a per-head distortion curve—the best achievable reconstruction error at a given rate jointly over r and b—and the global problem is to distribute total rate across heads to minimize the sum of distortions. This is the setting in which information-theoretic water-filling applies [14]. The resulting first-order conditions yield a per-head rule for splitting the rate between r and b and a marginal-distortion-equalizing schedule across heads, both solvable by a few iterations at calibration time.

We instantiate this view as **KV-COBRA** (*Co-Optimized Bit-Rank Allocation*), a two-level optimizer that runs entirely at calibration time. The inner level (**C1**) selects each head's $(r^\star, b^\star)$ by equating the marginal costs of truncation and quantization on its eigenvalue spectrum (Figure 1a). The outer level (**C2**) redistributes the total budget across heads via marginal-slope equalization. A Hadamard rotation fused into the SVD basis [11, 15] equalizes the steeply non-uniform per-channel variance of the retained coordinates at zero cost (Figure 1c). To make the objective query-aware rather than purely reconstruction-driven, we derive a second-order attention-KL bound that weights each direction by how much the quantizer hurts it and how much the attention map cares about it; reordering the basis by this weight before running C1/C2 makes the allocation query-aware. KV-COBRA reuses the same compression kernel as prior methods; what changes is how the bit-budget is allocated.

Across LLaMA-3.1-8B, Mistral-7B-v0.3, and Qwen2.5-7B-Instruct, KV-COBRA matches or beats every prior SVD- and rotation-based compressor at 0.5–2 bpd on Wikitext-2/PTB/C4 perplexity, LongBench F1, and zero-shot reasoning, and the gap widens at the lowest rates where allocation matters most. The takeaway is one line: *the bottleneck of low-bit KV compression is allocation across heads, not the kernel itself.*

Our contributions can be summarized as follows:

1. **A joint rank-bit optimizer with optimality conditions** (Section 3). C1 selects each head's (r, b) optimum, and C2 redistributes the total budget across heads. Both are gradient-free, run at calibration time, and add no per-token overhead.
2. **A query-aware upgrade via attention-KL reordering** (Section 4). A second-order bound on attention-KL yields a per-direction importance weight. Reordering the SVD basis by this weight before C1/C2 makes the solver optimize attention quality rather than reconstruction MSE.
3. **Graceful sub-1 bpd compression with no new kernel** (Section 6). KV-COBRA stays accurate down to 0.5 bpd where uniform-allocation baselines collapse, and the same allocator extends to joint $K+V$ compression (Tables 4, 5).

2 Related work

KV cache quantization. Existing KV-cache quantization methods reduce memory footprint through improved quantization and reconstruction. KIVI [6] quantizes K per-channel and V per-token with grouped bit widths; KVQuant [7] adds outlier handling and pre-RoPE [16] key quantization; GEAR [8] supplements grouped uniform quantization with a power-iteration low-rank residual. AQUA-KV [17] predicts each layer's cache from the previous one and quantizes the residual at 2–2.5 bits per value. All of these methods operate at a uniform per-head bit budget.

Table 1: **Allocation taxonomy for KV-cache compressors.** ✓ = optimized, fix = hyperparameter, eq. = equalized by rotation, — = N/A. $\Delta\mathrm{bpd}$ is the per-(channel, token) metadata overhead (Appendix E).

Method	Basis	layer	head	channel	rank	$\Delta\mathrm{bpd}$
KIVI [6]	identity	fix	fix	fix	—	$+1.0$
KVQuant [7]	identity	fix	fix	fix	—	$+0.16$
GEAR [8]	identity	fix	fix	fix	—	$+1.0$
TurboQuant [11]	rand. rotation	fix	fix	eq.	—	0
QuaRot [15]	Hadamard (fused)	fix	fix	eq.	—	0
SVDq [9]	per-layer SVD	fix	fix	fix	fix	0
KQ-SVD [10]	query-aware SVD	✓	fix	—	adaptive (ε)	0
KV-COBRA$_{\mathsf{KL}}$ (ours)	per-head SVD + Had.	✓	✓	eq.	✓	0

Rotation-based quantization. A second family first projects the cache into a basis that concentrates variance before quantizing: SVDq [9] uses a per-layer SVD basis with an 8-group mixed-precision schedule; TurboQuant [11] applies a random rotation plus per-channel Lloyd–Max quantization [18, 19]; QuaRot [15] fuses Hadamard rotations into the weights. KQ-SVD [10] instead uses a query-aware basis derived from the joint K/Q statistics. All fix rank and per-head budget as hyperparameters. Random-rotation scalar quantization with distortion guarantees predates KV compression, with EDEN [20] for distributed mean estimation and RaBitQ [21] for nearest-neighbor search. KV-COBRA adopts this standard kernel unchanged. What it adds is the allocation on top.

Eviction. A complementary line of work reduces cache size by discarding entries rather than compressing them, using importance scores such as accumulated attention (H2O [3]) or context-reconstruction loss (KVzip [22]). These methods operate on the token axis and compose with per-token compressors.

Architectural KV-cache reduction. Grouped-query attention [23], multi-head latent attention [24], cross-layer attention [25], and dynamic memory compression [26] share, merge, or compress KV at training time, whereas KV-COBRA is post-training, drop-in.

Allocation taxonomy. Table 1 organizes prior work by which allocation axes each method addresses; to our knowledge no prior method optimizes more than one. Throughout the paper we plot the *effective bpd* $\mathrm{bpd}_{\text{eff}} = \mathrm{bpd}_{\text{nominal}} + \Delta_{\text{method}}$ from the Δ column (Appendix E).

3 KV-COBRA under reconstruction MSE

This section derives the two-level bit-allocation solver under L_2 reconstruction MSE (Figure 2). Section 4 then upgrades the objective to attention-KL, making the solver query-aware.

Setup and notation. We consider a transformer with L layers, H KV heads with head dimension d. Given a context of length T, the model requires storing $2LHTd$ values as its KV cache. We denote by $K_{\ell,h}, V_{\ell,h} \in \mathbb{R}^{T\times d}$ the per-(layer, head) K/V matrices, by $\lambda_{\ell,h,1} \geq \cdots \geq \lambda_{\ell,h,d}$ the descending eigenvalues of their centered covariances, and by $U_{\ell,h} \in \mathbb{R}^{d\times d}$ the corresponding SVD eigenbases. We drop the side label when the argument applies to both K and V. We write bpd for the average bits per dimension and $B_{\mathrm{tot}} = LHd\,\mathrm{bpd}$ for the total bit budget.

Distortion model. KV-COBRA's Hadamard rotation mixes the retained SVD coordinates, pushing each channel toward a Gaussian marginal. For a zero-mean Gaussian channel with variance σ^2, a b-bit uniform scalar quantizer has MSE $\frac{\sigma^2}{12}2^{-2b} + o(2^{-2b})$ (Bennett [27]; Appendix A.1). We adopt the leading-order distortion

$$q(b) \;\coloneqq\; \tfrac{1}{12}\cdot 2^{-2b}, \tag{1}$$

However, a single fixed transform does not guarantee Gaussianity, and known bounds are asymptotic in dimension [28]. We therefore treat Gaussianity as a working assumption and validate it empirically at our head dimensions (Appendix A.3).

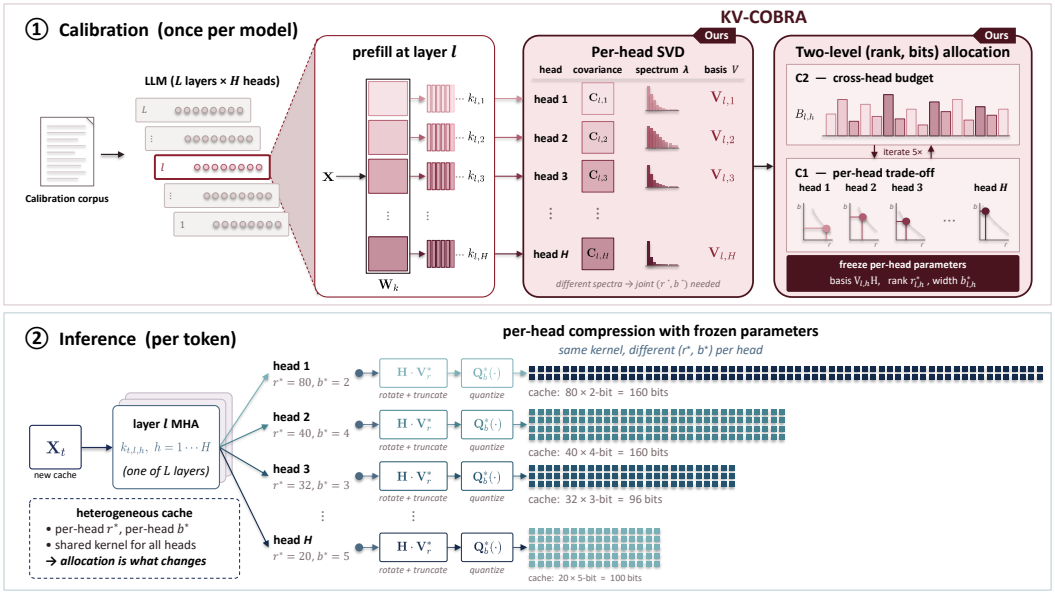


Figure 2: KV-COBRA pipeline. **Calibration.** Per-head SVD of prefill keys. A two-level C1+C2 allocator freezes $(V_{\ell,h}, r^\star_h, b^\star_h)$. **Inference.** Each key uses $H_{r^\star} \cdot V_{\ell,h}^\top$ and a $b^\star_h$-bit quantizer.

Projecting a head onto its top-r SVD directions discards variance $\sum_{i>r} \lambda_i$ (Eckart–Young [29]) and quantizing the r retained coordinates at $\bar{b}$ bits adds $q(\bar{b}) \sum_{i \le r} \lambda_i$. At the per-head level the compressor's effect on generation quality can be modeled by two natural distortion choices — the reconstruction MSE in the SVD basis, or the attention-KL divergence between clean and compressed softmax distributions — which both collapse to a common closed form

$$D(r, \bar{b}) \;=\; \underbrace{\sum_{i>r} w_i}_{\text{projection loss}} \;+\; \underbrace{q(\bar{b}) \sum_{i \le r} w_i}_{\text{quantization loss}}, \qquad w_i = \begin{cases} \lambda_i & \text{(MSE objective)}, \\ \sigma^2_{Q,i}\, \sigma^2_{K,i} & \text{(attention-KL objective)}, \end{cases} \tag{2}$$

differing only in whether each direction is weighted by the key eigenvalue or the query–key variance product (derivation of the KL form in Section 4). Both weightings plug into the same two-level C1/C2 solver; this section treats the MSE case (simpler, query-free), Section 4 upgrades to KL.

3.1 Problem formulation

KV-COBRA jointly optimizes the retained rank and bit budget of each head, rather than fixing them as global hyperparameters. Figure 1 illustrates this trade-off for LLaMA-3.1-8B. We use the MSE distortion D^{MSE} of Eq. (2); the KL formulation in Section 4 differs only in replacing $\lambda_{\ell,h,i} \mapsto w_{\ell,h,i}$. The resulting optimization problem is

$$\min_{\{(r_{\ell,h}, \bar{b}_{\ell,h})\}} \sum_{\ell,h} D_{\ell,h}\big(r_{\ell,h},\, \bar{b}_{\ell,h}\big) \qquad \text{s.t.} \qquad \sum_{\ell,h} r_{\ell,h}\, \bar{b}_{\ell,h} \le B_{\mathrm{tot}}, \tag{3}$$

with $D_{\ell,h}$ from Eq. (2) (the MSE choice). Problem (3) has a natural two-level decomposition:

Inner, per head (C1).
Given the head's budget $B_{\ell,h}$, jointly choose the retained rank $r_{\ell,h}$ and bit width $\bar{b}_{\ell,h} = B_{\ell,h}/r_{\ell,h}$ that minimize $D_{\ell,h}$.

Outer, across heads (C2).
Given the total budget B_{tot}, distribute it across heads as $\{B_{\ell,h}\}$ so that the total distortion $\sum_{\ell,h} D^\star_{\ell,h}(B_{\ell,h})$ is minimized.

Both levels admit first-order optimality conditions (Propositions B.1, C.2); we solve C1 by integer enumeration and C2 by a damped iteration that converges within five rounds on every tested model.

The compression kernel is inherited from the rotate-and-quantize family [9, 11], so with both levers frozen KV-COBRA reduces to a basic SVD pipeline. Sections 3.2–3.4 derive each level in turn.

3.2 Per-head optimum: rank and bit width together

Fix a single (layer, head). With a budget $r \cdot \bar{b} \le B$, increasing r reduces the projection loss $\sum_{i>r} \lambda_i$ but spreads the budget thinner; the optimum balances the two. Substituting $\bar{b} = B/r$ in Eq. (2) reduces D to a one-dimensional discrete objective $f(r) \coloneqq D(r, B/r)$, and on the even-integer rank grid (step 2) the forward difference is

$$\Delta f(r) = -(\lambda_{r+1}+\lambda_{r+2})\big(1 - q\big(\tfrac{B}{r+2}\big)\big) + \big(q\big(\tfrac{B}{r+2}\big) - q\big(\tfrac{B}{r}\big)\big) \sum_{i \le r} \lambda_i, \tag{4}$$

with any integer optimum $r^\star$ satisfying $\Delta f(r^\star - 2) \le 0 \le \Delta f(r^\star)$ (Proposition B.1): the projection-loss gain from two more directions matches the quantization-loss cost on the retained ones. We solve by enumerating $r \in \{2, 4, \ldots, \min(d, \lfloor B/b_{\min} \rfloor)\}$, setting $\bar{b} = \mathrm{round}(B/r)$ clipped to $[2, 8]$, and returning the $(r, \bar{b})$ with the smallest D — an $O(d/2)$ search per head, calibration-only. On typical calibration runs at 2 bpd, $r^\star$ spans $\sim[16, 100]$ and $b^\star \in [2, 5]$ across heads (Fig. 3).

3.3 Cross-layer redistribution: sum-distortion minimization

Heads differ: heavy-tailed ones saturate their budget early, flat ones stay starved. C2 uses C1 as a black-box oracle $D^\star_{\ell,h}(B_{\ell,h}) \coloneqq \min_{rb \le B_{\ell,h}} D_{\ell,h}(r, b)$ and redistributes so that every head sits at the same marginal distortion:

$$\min_{\{B_{\ell,h}\}} \sum_{\ell,h} D^\star_{\ell,h}(B_{\ell,h}) \quad \text{s.t.} \quad \sum_{\ell,h} B_{\ell,h} = B_{\mathrm{tot}}, \;\; B_{\ell,h} \ge B_{\min}. \tag{5}$$

Under standard convexity assumptions on $D^\star_{\ell,h}$ (Appendix C, Assumption C.1), the Lagrangian stationarity reduces to $(D^\star_{\ell,h})'(B_{\ell,h}) = -\nu$, i.e., marginal distortions equalize at the optimum. We solve (5) with a damped proportional update on $B_{\ell,h}$ (Appendix C). An exact DP solver attains the same PPL within noise but at substantially higher cost (Appendix D); we adopt the damped update.

3.4 Per-channel equalization via Hadamard rotation

The r retained SVD coordinates have steeply decaying variances (up to $10^3\times$; Figure 1c), so a per-channel quantizer would need an $L\times H$ grid of step-size schedules. We instead fuse a Hadamard rotation $V_r \to H_r V_r$, with H_r the $r\times r$ Walsh–Hadamard matrix and a fixed random sign flip drawn once at calibration and shared across all tokens [11, 15]. Absorbed into the same matrix–vector multiply (*zero latency*), it equalizes per-channel variance — the $10^3\times$ spread collapses to $O(1)$, so a single uniform $\bar{b}$-bit quantizer suffices — and pushes each channel toward Gaussian, the regime where Bennett's model (Appendix A.1) is tightest (validated in Appendix A.3).

Table 2: KV-COBRA pipeline: one-time calibration (1–6), then per-token inference (7–8).

Calibration (once per model; input: calibration data, target bpd)	
1 Per-head K covariance $\to \{\lambda_{\ell,h,i}\}, \{V_{\ell,h}\}$; query variance $\{\sigma^2_{Q,\ell,h,i}\}$ in the SVD basis	
2 $w_{\ell,h,i} \leftarrow \sigma^2_{Q,\ell,h,i} \cdot \lambda_{\ell,h,i}$; reorder $V_{\ell,h}$ by $w_{\ell,h,i}$ descending	[§4]
3 $B_{\ell,h} \leftarrow \mathrm{bpd} \cdot d$ for all (ℓ, h)	
4 **5 rounds:** $(r^\star_{\ell,h}, \bar{b}^\star_{\ell,h}) \leftarrow$ C1$(w_{\ell,h}, B_{\ell,h})$; $B_{\ell,h} \leftarrow$ C2-update	[§3.2, §3.3]
5 Fuse Hadamard: $V_{\ell,h} \leftarrow H_{r^\star} \cdot V_{\ell,h}[:, : r^\star]$	[§3.4]
6 Store $\{V_{\ell,h},\ r^\star_{\ell,h},\ \bar{b}^\star_{\ell,h}\}$ per head	
Inference (per token, per head)	
7 $\hat{z}_t \leftarrow \mathrm{quantize}(V_{\ell,h}^\top K_t,\ \bar{b}^\star_{\ell,h})$	[project+Hadamard as one multiply]
8 Cache $\hat{z}_t$; dequantize on attention read	

4 KV-COBRA under attention-KL

The L_2 formulation of Section 3 minimizes reconstruction error in the SVD basis, but downstream quality depends on what the attention layer *does* with the cache, not on the cache itself. The L_2 objective weights every direction by its K-eigenvalue λ_i, regardless of whether the query cares about it. Write the quantized key as $\tilde{K}_t = K_t + E_t$, the clean attention distribution as $p = \mathrm{softmax}(s)$ with logits $s_t = \langle Q, K_t \rangle / \sqrt{d}$, and the quantized distribution as $\tilde{p} = \mathrm{softmax}(s + \Delta s)$.

Proposition 4.1 (Attention-KL upper bound)**.** *Under the high-resolution additive-noise model (*$\mathbb{E}[E_t]$=0*, channel-wise uncorrelated with* $\mathrm{Var}[E_{t,i}]$=$q(\bar{b})\,\sigma^2_{K,i}$*, $Q \perp E_t$),*

$$\mathbb{E}\big[\mathrm{KL}(p \,\|\, \tilde{p})\big] \underset{(a)}{\leq} \tfrac{1}{2}\,\mathbb{E}\big[\Delta s^\top F(p)\,\Delta s\big] \underset{(b)}{\leq} \tfrac{1}{4}\sum_{t=1}^{T} \mathrm{Var}[\Delta s_t] \underset{(c)}{=} C \sum_{i=1}^{r} \underbrace{\sigma^2_{Q,i}\,\sigma^2_{K,i}}_{=:\,w_i}\, q(\bar{b}), \qquad (6)$$

with $C = T/(4d)$. The relevant per-direction quality weight is therefore $w_i = \sigma^2_{Q,i}\sigma^2_{K,i}$, not the K-spectrum λ_i alone.

(a) Taylor expansion of softmax KL, with $F(p) = \mathrm{diag}(p) - pp^\top$ the softmax Fisher information. Used as a leading-order proxy; the cubic remainder is signed and absorbed into our weight ranking, not the absolute KL value.

(b) Fisher tight bound. $F(p) \preceq \frac{1}{2} I$ (Popoviciu on unit vectors), giving the $\frac{1}{4}$ leading constant after combining with the Taylor $\frac{1}{2}$.

(c) Query–key factorization. $\Delta s_t = Q \cdot E_t^\top / \sqrt{d}$; assuming $Q \perp E_t$ and channel-wise uncorrelated ε_i gives $\mathrm{Var}[\Delta s_t] = d^{-1} \sum_i \mathbb{E}[Q_i^2]\,\mathbb{E}[\varepsilon_i^2]$, and with $\mathbb{E}[Q_i^2] =: \sigma^2_{Q,i}$ this yields the per-direction weight $w_i = \sigma^2_{Q,i}\sigma^2_{K,i}$. Summing over the T tokens absorbs a factor of T into C.

Reordering and plug-in. The weights w_i are *not* monotone in λ_i: a direction with small λ_i but large $\sigma^2_{Q,i}$ would be discarded by an MSE-ordered truncation the query actually attends to. Before running C1/C2 we reorder the SVD basis by w_i descending (once at calibration, negligible cost), so "top-r" means the r directions with the largest attention-KL importance. These reordered weights replace λ_i in D^{KL} of Eq. (2), and the forward difference of Eq. (4) and the cross-head sum-distortion envelope of Eq. (5) carry over verbatim. The resulting method is KV-COBRA$_{\mathsf{KL}}$ (main); the L_2 version serves as ablation. Table 2 summarizes. The full C1+C2 calibration takes under 200 ms across all three models (Appendix D), comparable to generating a few dozen tokens at standard FP16 decode rates.

5 Experimental setup

Scope. We compress K pre-RoPE [16]; post-RoPE the per-head covariance is position-dependent and no static SVD basis captures it (Appendix G). The body focuses on k-only because the V spectrum is structurally flatter (Appendix F); joint $K{+}V$ results follow in Tables 4 and 5.

Models. Three independently trained 7–8B grouped-query backbones: LLaMA-3.1-8B [30], Mistral-7B-v0.3 [31], and Qwen2.5-7B-Instruct [32], all loaded in `fp16`.

Benchmarks. (i) Perplexity on Wikitext-2 [33], PTB [34], and C4 [35] (sliding window 2,048, 32k evaluation tokens per dataset); (ii) zero-shot accuracy on ARC-c [36], HellaSwag [37], PIQA [38], WinoGrande [39], MMLU [40] (200 samples per task); (iii) LongBench [41] F1 on five tasks (NarrativeQA [42], QASPER [43], MultiFieldQA-en, HotpotQA [44], MuSiQue [45]; 50 samples per task).

Baselines and protocol. We compare against KIVI [6], KVQuant [7], GEAR [8], TurboQuant [11], SVDq [9], and KQ-SVD [10]. The pipeline (calibration, compression, evaluation) is deterministic given the seed; multi-seed cells (Table 3, Fig. 4) report mean $\pm$ 1 SD over 5 seeds. Per-baseline bit-width quirks, calibration details, and the full bit-rate scan are in Appendix H.

6 Results

Per-head $(r^\star, b^\star)$ optima spread across many operating points and shift with budget and model (Fig. 3). A single fixed (r, b) cannot fit all heads, so per-head allocation is the axis on which the methods below differ. C2 then equalizes per-head distortion across heads. With uniform per-head budgets a few heads carry most of the residual error, while the C2 allocation flattens the curve and reduces the cross-head distortion std by 2–8×.

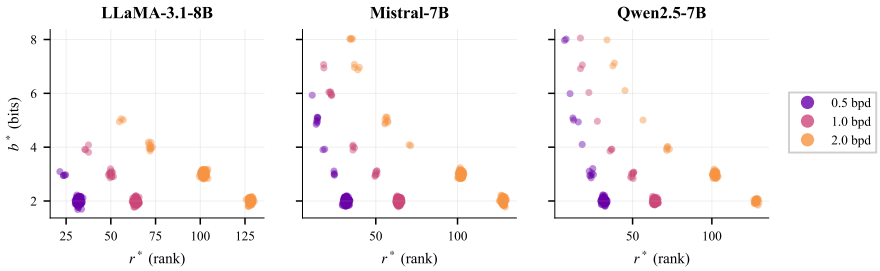


Figure 3: Per-head $(r^\star, b^\star)$ selected by C1 across three models, colored by target bpd (0.5, 1.0, 2.0).

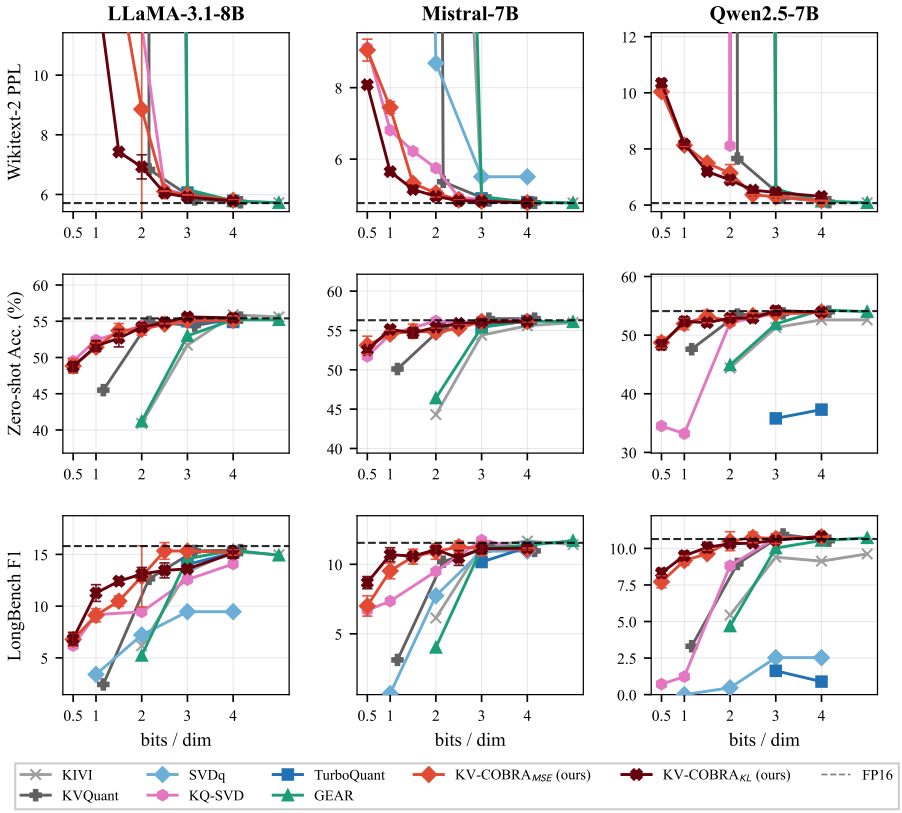


Figure 4: Main k-only results. Dashed lines mark FP16; error bars are ± 1 SD over 5 seeds.

Main comparison. Figure 4 sweeps the full 0.5–4 bpd range across KIVI, KVQuant, GEAR, TurboQuant, SVDq, and KQ-SVD. At $\geq$2 bpd the choice of allocator barely matters. At low bpd KV-COBRA$_{\mathsf{KL}}$ is the most graceful overall, holding within range on all three metric averages. KIVI and GEAR are integer-only and cannot enter the regime, while SVDq and KQ-SVD degrade sharply: at 1 bpd KV-COBRA$_{\mathsf{KL}}$ nearly doubles the LongBench F1 of integer-only baselines. Table 3 zooms in on 0.5 and 1.0 bpd, where PPL is the mean over Wikitext-2, PTB, and C4 and zero-shot and LongBench are each the mean over five tasks (multi-seed averages, Appendix H). KV-COBRA$_{\mathsf{MSE}}$

wins on PPL (its direct objective), while KV-COBRA$_{\mathsf{KL}}$ wins on LongBench and matches KV-COBRA$_{\mathsf{MSE}}$ on zero-shot — the split predicted by the framing (PPL is reconstruction-aligned, LongBench attention-aligned).

Table 3: Metric-group means at 0.5 and 1.0 bpd in k-only mode across the three models. PPL is the mean over Wikitext-2, PTB, and C4 (5 seeds). Zero-shot and LongBench are each the mean over five tasks (5 seeds). **Bold** marks the better of KV-COBRA$_{\mathsf{MSE}}$ and KV-COBRA$_{\mathsf{KL}}$ per cell. Per-task breakdown is in Appendix I.

	0.5 bpd						1.0 bpd						FP16		
	LL		Mi		Qw		LL		Mi		Qw		LL	Mi	Qw
Metric	MSE	KL	MSE	KL	MSE	KL	MSE	KL	MSE	KL	MSE	KL			
PPL (↓)	**31.17** ± 2.21	37.31 ± 6.44	36.50 ± 1.46	**36.24** ± 0.68	**16.33** ± 0.15	17.17 ± 0.23	24.76 ± 1.14	**20.26** ± 0.87	27.18 ± 0.97	**15.90** ± 0.13	**12.97** ± 0.09	13.20 ± 0.06	7.67	13.15	9.50
Zero-shot (↑)	**48.8** ± 1.0	48.5 ± 0.5	**53.4** ± 0.9	52.5 ± 0.3	**48.7** ± 0.9	48.5 ± 0.7	51.5 ± 0.6	**51.6** ± 0.7	54.6 ± 0.5	**55.1** ± 0.5	51.7 ± 0.7	**52.5** ± 0.7	55.4	56.3	54.1
LongBench (↑)	**6.83** ± 0.63	6.75 ± 0.58	7.02 ± 0.73	**8.73** ± 0.42	7.69 ± 0.40	**8.30** ± 0.31	9.25 ± 0.54	**11.36** ± 0.77	9.55 ± 0.56	**10.72** ± 0.49	9.10 ± 0.29	**9.53** ± 0.30	15.80	11.54	10.64

Joint K+V compression. We hold the total budget $T = K_{\text{bpd}} + V_{\text{bpd}}$ fixed and sweep both sides (Tables 4, 5). The V side uses a uniform per-head budget because its spectrum is too flat for cross-head redistribution to help (Appendix F). On PPL, V-heavy splits dominate and KV-COBRA$_{\mathsf{KL}}$ takes the per-block optimum on 7 of 9 (model, T) blocks; conversely, V_{bpd}=2 collapses on LLaMA and Mistral across all three methods (PPL >150). KQ-SVD's failure on Qwen's K_{bpd}=1 cells (PPL $\sim 10^4$) reflects its halved KV-head count without per-head allocation. Table 5 reports the same sweep on LongBench F1.

Table 4: K+V joint compression on Wikitext-2 PPL (↓). Each row scans splits $(K_{\text{bpd}}, V_{\text{bpd}})$ at fixed total T. Both sides are compressed (V uniform per-head, Appendix F). The FP16 column is the uncompressed reference. Shaded cell marks the best PPL within each (model, T) block.

		$T=5$			$T=6$				$T=7$					
	Method	$(1,4)$	$(2,3)$	$(3,2)$	$(1,5)$	$(2,4)$	$(3,3)$	$(4,2)$	$(1,6)$	$(2,5)$	$(3,4)$	$(4,3)$	$(5,2)$	FP16
LLaMA-3.1-8B	KV-COBRA$_{\mathsf{MSE}}$	17.35	38.84	732.1	15.79	17.88	19.17	550.4	15.21	15.48	6.69	17.26	505.7	
	KV-COBRA$_{\mathsf{KL}}$	14.63	24.79	490.7	13.54	8.86	16.55	465.1	13.35	7.74	6.52	16.80	541.3	5.71
	KQ-SVD	26.14	50.82	528.1	15.19	15.01	27.36	559.5	13.31	12.50	6.83	25.21	551.7	
Mistral-7B	KV-COBRA$_{\mathsf{MSE}}$	7.96	8.59	285.7	7.43	5.46	7.92	326.3	7.35	5.25	4.99	7.92	311.3	
	KV-COBRA$_{\mathsf{KL}}$	5.97	10.04	252.6	5.69	5.31	8.00	305.9	5.65	5.10	4.98	7.89	357.1	4.77
	KQ-SVD	8.14	19.76	184.1	7.04	6.42	10.48	162.1	6.80	5.85	5.10	9.59	159.6	
Qwen2.5-7B	KV-COBRA$_{\mathsf{MSE}}$	8.29	8.47	12.86	8.16	7.78	6.92	11.75	8.19	7.60	6.43	6.78	11.57	
	KV-COBRA$_{\mathsf{KL}}$	8.39	7.80	16.62	8.25	7.26	7.65	13.56	8.22	7.20	7.11	7.05	12.13	6.07
	KQ-SVD	11137.	10.04	19.60	10812.	8.35	7.37	17.51	10932.	8.12	6.47	7.21	17.25	

Table 5: K+V joint compression, LongBench F1 (↑, mean over NarrativeQA, QASPER, MultiFieldQA-en, HotpotQA, and MuSiQue at 50 samples per task). Shaded cell marks the best F1 within each (model, T) block.

		$T=5$			$T=6$				$T=7$					
	Method	$(1,4)$	$(2,3)$	$(3,2)$	$(1,5)$	$(2,4)$	$(3,3)$	$(4,2)$	$(1,6)$	$(2,5)$	$(3,4)$	$(4,3)$	$(5,2)$	FP16
LLaMA-3.1-8B	KV-COBRA$_{\mathsf{MSE}}$	8.64	6.75	3.25	9.86	8.31	9.71	3.73	8.08	8.89	13.59	8.98	3.66	
	KV-COBRA$_{\mathsf{KL}}$	11.06	7.78	2.81	10.67	11.25	8.62	3.32	12.16	12.10	12.35	8.95	3.19	15.80
	KQ-SVD	6.21	6.63	3.59	8.23	8.14	7.02	3.42	8.78	9.23	11.55	7.66	3.38	
Mistral-7B	KV-COBRA$_{\mathsf{MSE}}$	8.42	9.07	1.60	9.71	10.25	9.13	0.95	9.35	10.80	11.16	9.62	1.01	
	KV-COBRA$_{\mathsf{KL}}$	11.13	8.41	1.95	10.04	10.64	9.83	1.12	10.73	10.22	9.90	9.39	1.20	11.54
	KQ-SVD	6.90	5.48	1.51	8.50	9.51	8.21	1.84	8.04	10.34	10.68	9.11	2.20	
Qwen2.5-7B	KV-COBRA$_{\mathsf{MSE}}$	9.23	9.78	8.39	8.99	9.41	10.35	9.24	8.89	9.36	9.82	10.16	9.07	
	KV-COBRA$_{\mathsf{KL}}$	9.85	10.00	8.25	8.60	10.27	10.04	8.52	9.19	10.32	10.48	9.84	9.35	10.64
	KQ-SVD	1.16	8.03	7.29	1.17	8.70	10.05	9.06	1.28	8.78	10.10	10.30	8.59	

Additional experiments. Appendix I reports the full per-dataset PPL and per-task zero-shot/LongBench breakdown (Tables A4 and A5) and a per-head attention-primitive wall-clock measurement showing KV-COBRA is 1.79–$1.89\times$ faster than FP16-SDPA at $r{=}16$ across $T \in \{32\text{k}, 64\text{k}, 128\text{k}\}$ (Table A6). Appendix G covers pre- versus post-RoPE, and Appendix J adds RULER long-context retrieval results.

7 Discussion and limitations

Allocation is the bottleneck, not the kernel. KV-COBRA reuses the same SVD-rotate-and-quantize kernel as prior work, and every gain we report comes from redirecting the same bits to better places. The C2 surrogate-optimality check (Appendix D) reinforces this point. Replacing our damped proportional update with an exact DP solver for the sum-distortion minimization reduces $\sum D^\star$ by $\sim$ 12–16% but actually *degrades* PPL by ~ 0.5 on average. The surrogate itself is the proxy at fault, not the solver — which is exactly why the attention-KL upgrade (Sec. 4) helps where any MSE-side refinement cannot. Appendix D pins the fault down by replacing the Bennett estimate with measured per-head distortion, after which the same exact solver improves on the shipped allocator. For practitioners, this means *where* bits are spent — which head, which direction — matters more at low bpd than how they are quantized.

When does the KL upgrade pay off? At $\geq$2 bpd L_2 and attention-KL agree within noise. Below 1 bpd, KL typically improves LongBench F1 even when MSE wins on PPL (Table 3). KL also takes the per-block optimum on 7 of the 9 (model, T) blocks in the K+V joint sweep (Table 4).

Asymmetric K/V budgets. V-heavy splits dominate the joint $K{+}V$ sweep (Tables 4, 5), reflecting V's flatter spectrum (Appendix F). The same $(r^\star, b^\star)$ allocator handles both sides without any K-specific assumption, generalizing the per-head principle to the asymmetric regime. Appendix I breaks the asymmetry down per task (Table A7).

Limitations.

- **Scope.** Our comparison is restricted to per-token compressors on the same (r, b) axes. Two orthogonal families fall outside this scope: token-axis eviction (H2O [3], StreamingLLM [5]), which discards entries rather than compressing them, and entropy coding on top of a quantizer. Both compose with KV-COBRA in principle, but a fused evaluation is left for follow-up.
- **Wall-clock.** We do not report end-to-end latency, but Appendix I reports primitive-level attention speedup over FP16-SDPA.
- **Calibration stationarity.** $(r^\star, b^\star)$ and budgets are computed once from 32 Wikitext-2 samples. The allocation is insensitive to the calibration corpus and size and transfers across domains (Appendix I), but refresh-during-generation is not studied.
- **Distortion model.** Bennett's leading-order $q(b)$ [27] assumes near-Gaussian channels. A single fixed Hadamard transform carries no distributional guarantee [28], so we rely on the empirical validation of Appendix A.3. The finite-rate correction at $b{=}1$ can alter the rank selected by C1 (Appendix A.1).

8 Conclusion

Every rotate-and-quantize KV-cache compressor hides two allocation decisions — per-head rank and per-head bit allocation — that prior work treats as fixed hyperparameters. **KV-COBRA** (*Co-Optimized Bit-Rank Allocation*) solves them jointly through a two-level decomposition, a per-head (r, b) optimum (C1) and a cross-head budget distributor (C2), and adds an attention-KL reordering that makes both solvers query-aware. The pipeline is gradient-free and calibration-only, and across LLaMA-3.1-8B, Mistral-7B, and Qwen2.5-7B it shows the smallest accuracy degradation among evaluated methods at low bpd. The same allocator extends to the joint $K{+}V$ regime, where V-heavy splits typically dominate. The takeaway is that bit allocation, not the quantization kernel, is where low-bpd gains live, and stronger rotations or entropy coding can layer on top without changing the allocator. Natural extensions include MLA and MoE architectures, refining the V side, and composing with token-axis eviction policies.

Acknowledgments and Disclosure of Funding

This work was supported in part by a grant of Korean ARPA-H Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (Grant number: RS-2025-25454431) and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) in 2026 (RS-2024-00453301, MASC: Convergence of Multimodal AI and Semantic Communications for Hyper-Connected Hyper-Intelligent Public Safety Systems, 50 %).

References

[1] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

[2] Zhihang Yuan, Yuzhang Shang, Yang Zhou, Zhen Dong, Zhe Zhou, Chenhao Xue, Bingzhe Wu, Zhikai Li, Qingyi Gu, Yong Jae Lee, et al. Llm inference unveiled: Survey and roofline model insights. *arXiv preprint arXiv:2402.16363*, 2024.

[3] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

[4] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

[5] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[6] 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. *Proceedings of Machine Learning Research*, 235:32332–32344, 2024.

[7] Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

[8] Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. Gear: An efficient kv cache compression recipe for near-lossless generative inference of llm. *arXiv preprint arXiv:2403.05527*, 2024.

[9] Yankun Hong, Xing Li, Hui-Ling Zhen, Xianzhi Yu, Wulong Liu, and Mingxuan Yuan. SVDq: 1.25-bit and 410x key cache compression for LLM attention. *arXiv preprint arXiv:2502.15304*, 2025.

[10] Damien Lesens, Beheshteh T Rakhshan, and Guillaume Rabusseau. Kq-svd: Compressing the kv cache with provable guarantees on attention fidelity. *arXiv preprint arXiv:2512.05916*, 2025.

[11] Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. Turboquant: Online vector quantization with near-optimal distortion rate. *arXiv preprint arXiv:2504.19874*, 2025.

[12] Claude E Shannon et al. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163):1, 1959.

[13] Toby Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, 1971.

[14] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[15] 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. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

[16] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[17] Alina Shutova, Vladimir Malinovskii, Vage Egiazarian, Denis Kuznedelev, Denis Mazur, Nikita Surkov, Ivan Ermakov, and Dan Alistarh. Cache me if you must: Adaptive key-value quantization for large language models. *arXiv preprint arXiv:2501.19392*, 2025.

[18] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28 (2):129–137, 1982.

[19] Joel Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(1): 7–12, 1960.

[20] Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben Itzhak, and Michael Mitzenmacher. EDEN: Communication-efficient and robust distributed mean estimation for federated learning. In *International Conference on Machine Learning*, pages 21984–22014. PMLR, 2022.

[21] Jianyang Gao and Cheng Long. RaBitQ: Quantizing high-dimensional vectors with a theoretical error bound for approximate nearest neighbor search. *Proceedings of the ACM on Management of Data*, 2(3):1–27, 2024.

[22] Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-agnostic kv cache compression with context reconstruction. *arXiv preprint arXiv:2505.23416*, 2025.

[23] Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

[24] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[25] William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan-Kelley. Reducing transformer key-value cache size with cross-layer attention. *Advances in Neural Information Processing Systems*, 37:86927–86957, 2024.

[26] Piotr Nawrot, Adrian Łańcucki, Marcin Chochowski, David Tarjan, and Edoardo M Ponti. Dynamic memory compression: Retrofitting llms for accelerated inference. *arXiv preprint arXiv:2403.09636*, 2024.

[27] William Ralph Bennett. Spectra of quantized signals. *The Bell System Technical Journal*, 27(3): 446–472, 1948.

[28] Ran Ben-Basat, William Kuszmaul, Michael Mitzenmacher, Amit Portnoy, and Shay Vargaftik. Quantizing with randomized hadamard transforms: Efficient heuristic now proven. *arXiv preprint arXiv:2605.06014*, 2026.

[29] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[30] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[31] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

[32] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024.

[33] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[34] Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[36] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[37] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800, 2019.

[38] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

[39] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[40] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.

[41] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 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)*, pages 3119–3137, 2024.

[42] Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

[43] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, 2021.

[44] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380, 2018.

[45] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[46] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018. doi: 10.1017/9781108231596.

[47] Robert M. Gray and David L. Neuhoff. Quantization. *IEEE transactions on information theory*, 44(6):2325–2383, 1998.

[48] Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The quarterly journal of mathematics*, 11(1):50–59, 1960.

[49] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[50] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

A Distortion model and head-level $D(r, b)$

This section establishes the per-head distortion function $D(r, b)$ that the main paper uses without proof. We need two ingredients: (i) the per-channel quantization MSE $q(b)\cdot\sigma^2$ (Lemma A.1); and (ii) the Eckart–Young identity for the residual MSE of an orthogonal projection (Lemma A.2).

A.1 The uniform scalar quantizer after Hadamard rotation

KV-COBRA applies a Hadamard rotation to the retained SVD coordinates before quantization. The rotation equalizes per-channel variance to the average $\bar{\sigma}^2 = r^{-1}\sum_{i\le r}\lambda_i$: with the scaling $H_r H_r^\top = I$, Walsh–Hadamard rows have squared entries $1/r$, so for any diagonal covariance $\mathrm{diag}(\lambda_1, \ldots, \lambda_r)$ every rotated channel has variance $r^{-1}\sum_j \lambda_j$. Let $X \sim \mathcal{N}(0, \sigma^2)$ and let Q_b denote a uniform mid-tread quantizer at rate b bits, applied to X clipped to a range $[-L\sigma, L\sigma]$ with step $\Delta = 2L\sigma/2^b$ (the loading factor L is absorbed into α below). Bennett's high-resolution formula [27] then gives

$$\mathbb{E}\big[(X - Q_b(X))^2\big] = \alpha \cdot 2^{-2b} \cdot \sigma^2 + o(2^{-2b}), \tag{7}$$

where $\alpha > 0$ depends on the loading factor L (the standard normalization $\alpha = 1/12$ corresponds to matching the granular term $\Delta^2/12$). Since $D(r, b)$ is the sum of an α-free truncation term and an α-proportional quantization term under the product constraint $rb \le B$, the C1 and C2 optimizers only depend on α up to a ± 1 rank-grid shift that is negligible at $b \gtrsim 2$. We therefore normalize

$$q(b) \coloneqq \tfrac{1}{12} \cdot 2^{-2b}, \tag{8}$$

absorbing α into an overall scale. The Hadamard rotation produces each channel as a $\pm 1/\sqrt{r}$ weighted sum of r retained coordinates, the entries of $Y = H_r U_r^\top X$ are uncorrelated and, for moderately large r, approximately Gaussian [46]. We verify this empirically (Appendix A.3): post-rotation channels track $\mathcal{N}(0, 1)$ closely up to the tails.

Lemma A.1 (Per-channel quantization MSE)**.** *Let $X \sim \mathcal{N}(0, \sigma^2)$ be quantized by a uniform mid-tread quantizer at b bits. Under the normalization of* (8)*, $\mathbb{E}[(X - Q_b(X))^2] = q(b) \cdot \sigma^2 + o(2^{-2b})$.*

Proof. Standard Bennett high-resolution result [27, 47]; follows from partitioning $\mathbb{R}$ into cells of width Δ, Taylor-expanding the Gaussian density around each cell center, and summing the leading-order $\Delta^2/12$ contributions; absorbing the α prefactor into (8) gives the stated form. □

A.2 Truncation in the SVD basis

Lemma A.2 (Truncation distortion in the original basis; Eckart–Young 1936 [29], Mirsky 1960 [48])**.** *Let X be any zero-mean random vector in $\mathbb{R}^d$ with finite second moments, write its covariance as $\Sigma = U\Lambda U^\top$, and let $U_r \in \mathbb{R}^{d\times r}$ be the first r columns of U. Define $\hat{X} = U_r U_r^\top X$. Then*

$$\mathbb{E}\|X - \hat{X}\|^2 = \sum_{i>r} \lambda_i.$$

Proof. Rotate to the eigenbasis: $Y \coloneqq U^\top X$ has diagonal covariance Λ, so $\mathrm{Var}[Y_i] = \lambda_i$ with the Y_i uncorrelated. Projection onto the top-r directions zeros out $Y_{r+1}, \ldots, Y_d$, and U is an isometry, so $\|X - \hat{X}\|^2 = \sum_{i>r} Y_i^2$. Taking expectations, $\mathbb{E}\|X - \hat{X}\|^2 = \sum_{i>r}\lambda_i$. □

A.3 Empirical post-Hadamard Gaussianity

Bennett's formula and the high-resolution per-channel MSE assumption (Assumption A.3) require each post-Hadamard channel marginal to be approximately Gaussian. A single drawn-once sign pattern yields no distributional guarantee, and the known bounds for composed randomized Hadamard transforms vanish only asymptotically in the dimension [28], so at our retained ranks (r as small as 16) the residual is not negligible a priori. We therefore validate this on the calibration set across all $(\text{layer}, \text{head}, \text{channel})$ triples on three models: pre-rotation excess kurtosis is heavy-tailed (median $\kappa_{\text{pre}} \approx 0.30$), post-rotation it concentrates sharply around zero (median $|\kappa_{\text{post}}| < 0.20$, 95th percentile < 1.10; Fig. A1). Q–Q plots (Fig. A2) confirm the post-rotation channels track $\mathcal{N}(0, 1)$ where the pre-rotation channels show the standard heavy-tailed S-curve.

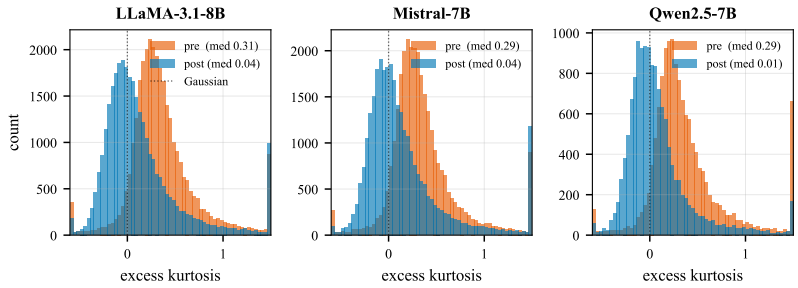


Figure A1: Excess kurtosis distribution across all (layer, head, channel) triples (one panel per model). Pre-Hadamard (orange) is heavy-tailed (median $\kappa \approx 0.30$); post-Hadamard (blue) collapses near zero (median $|\kappa| < 0.20$).

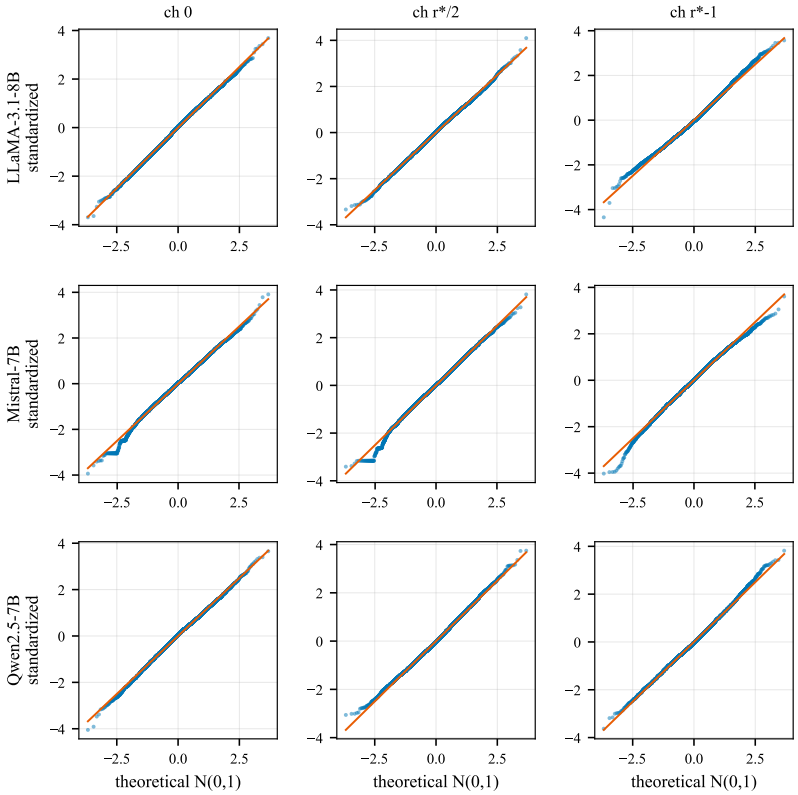


Figure A2: Q–Q grid against $\mathcal{N}(0,1)$. Rows: three models; columns: representative channels (first, mid, last) of one "median-behavior" head per model. Post-Hadamard samples track the diagonal up to $\pm 2.5\sigma$ with mild tail deviation, consistent with Bennett's high-resolution regime.

A.4 Putting it together: head-level $D(r, b)$

The full head-level distortion under the rotate-and-quantize pipeline (top-r SVD projection followed by per-channel uniform quantization) is the sum of the two independent contributions from Lemma A.2 (projection residual) and Lemma A.1 (uniform quantizer).

Assumption A.3 (High-resolution per-channel MSE)**.** Let $Y \in \mathbb{R}^r$ be the pre-quantization vector and $\varepsilon_i := Q_b(Y_i) - Y_i$ the per-channel quantization error. We assume the marginal MSE matches Bennett's high-resolution formula channel-wise: $\mathbb{E}[\varepsilon_i^2] = q(b) \cdot \mathrm{Var}(Y_i) + o(q(b))$ for every $i \in \{1, \ldots, r\}$. This holds exactly as $b \to \infty$ when Y_i is approximately $\mathcal{N}(0, \mathrm{Var}(Y_i))$; the finite-$b$ deviation is absorbed into the $o(q(b))$ remainder.

Lemma A.4 (Head-level reconstruction MSE)**.** *Let $X \in \mathbb{R}^d$ be zero-mean with covariance $\Sigma = U\Lambda U^\top$, eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$. Let U_r be the first r columns of U and $H_r \in \mathbb{R}^{r\times r}$ a scaled Walsh–Hadamard matrix ($H_r H_r^\top = I$). Define the compressed reconstruction*

$$\hat{X} \;=\; U_r H_r^\top \, Q_b^{\otimes}\big(H_r U_r^\top X\big),$$

where $Q_b^{\otimes}$ applies a scalar uniform mid-tread quantizer at rate b to each of its r inputs. Under Assumption A.3,

$$D(r,b) \;\coloneqq\; \mathbb{E}\|X-\hat{X}\|^2 = \sum_{i>r}\lambda_i \;+\; q(b)\sum_{i\leq r}\lambda_i \;+\; o\big(q(b)\big). \tag{9}$$

Proof. Step 1 (orthogonal decomposition). Write

$$X-\hat{X} = \underbrace{(I-U_rU_r^\top)X}_{=:\,\xi_1\;\in\,\mathrm{range}(U_r)^\perp} + \underbrace{U_rH_r^\top\big(H_rU_r^\top X - Q_b^{\otimes}(H_rU_r^\top X)\big)}_{=:\,\xi_2\;\in\,\mathrm{range}(U_r)}.$$

The two subspaces are orthogonal deterministically, so $\|X-\hat{X}\|^2 = \|\xi_1\|^2+\|\xi_2\|^2$ and $\mathbb{E}\|X-\hat{X}\|^2 = \mathbb{E}\|\xi_1\|^2 + \mathbb{E}\|\xi_2\|^2$.

Step 2 (truncation term). By Lemma A.2, $\mathbb{E}\|\xi_1\|^2 = \sum_{i>r}\lambda_i$.

Step 3 (quantization term). Let $Y \coloneqq H_rU_r^\top X \in \mathbb{R}^r$ and $\varepsilon \coloneqq Q_b^{\otimes}(Y) - Y$. Since $U_rH_r^\top$ has orthonormal columns, $\|\xi_2\|^2 = \|U_rH_r^\top\varepsilon\|^2 = \|\varepsilon\|^2 = \sum_{i=1}^r \varepsilon_i^2$ (Euclidean-norm definition, deterministic). By linearity of expectation and Assumption A.3,

$$\mathbb{E}\|\xi_2\|^2 = \sum_{i=1}^r \mathbb{E}[\varepsilon_i^2] = q(b)\sum_{i=1}^r \mathrm{Var}(Y_i) + o(q(b)).$$

Step 4 (variance equalization under Hadamard). Under the normalization $H_rH_r^\top = I$, Hadamard rows have squared entries $1/r$, so every diagonal entry of $\mathrm{Cov}(Y) = H_r\,\mathrm{diag}(\lambda_1,\ldots,\lambda_r)\,H_r^\top$ equals $\bar{\sigma}^2 = \frac{1}{r}\sum_{i\leq r}\lambda_i$, and

$$\sum_{i=1}^r \mathrm{Var}(Y_i) = r\bar{\sigma}^2 = \sum_{i\leq r}\lambda_i.$$

Step 5 (combine). Steps 1–4 yield $\mathbb{E}\|X-\hat{X}\|^2 = \sum_{i>r}\lambda_i + q(b)\sum_{i\leq r}\lambda_i + o(q(b))$. □

Eq. (9) is the same $D(r,b)$ that the main paper uses without proof. Everything about KV-COBRA's per-head decision is an optimization over this quantity.

B The per-head optimum (C1)

With $D(r,b)$ fully justified, we can now state the discrete-integer optimality conditions for the per-head joint (r,b) optimizer under a product budget constraint.

Proposition B.1 (Per-head discrete first-order condition)**.** *Let $D(r,b)$ be as in Lemma A.4, and consider the integer problem*

$$\min_{\substack{r\in\{r_{\min},r_{\min}+2,\ldots,d\}\\ b\in\{b_{\min},\ldots,b_{\max}\}}} D(r,b) \quad \textit{s.t.} \quad r\cdot b \leq B.$$

Substituting $b = B/r$ reduces this to the one-dimensional discrete objective $f(r) \coloneqq D(r,B/r)$, whose forward difference is

$$\Delta f(r) \;\coloneqq\; f(r{+}2)-f(r) \;=\; -\big(\lambda_{r+1}+\lambda_{r+2}\big)\big(1-q(B/(r{+}2))\big) \\ +\;\big(q(B/(r{+}2))-q(B/r)\big)\sum_{i\leq r}\lambda_i.$$

Any integer optimum $r^\star$ on this grid satisfies the two-sided first-order condition

$$\Delta f(r^\star-2) \;\leq\; 0 \;\leq\; \Delta f(r^\star),$$

with the convention $\Delta f(r_{\min}-2) \coloneqq -\infty$ and $\Delta f(d) \coloneqq +\infty$ at the boundary.

Proof. Compute

$$\begin{aligned} f(r{+}2) - f(r) &= \Big[\sum_{i>r+2} \lambda_i\Big] - \Big[\sum_{i>r} \lambda_i\Big] + q\big(\tfrac{B}{r+2}\big) \sum_{i\le r+2} \lambda_i - q\big(\tfrac{B}{r}\big) \sum_{i\le r} \lambda_i \\ &= -(\lambda_{r+1}{+}\lambda_{r+2}) + q\big(\tfrac{B}{r+2}\big)(\lambda_{r+1}{+}\lambda_{r+2}) \\ &\quad + \big(q\big(\tfrac{B}{r+2}\big) - q\big(\tfrac{B}{r}\big)\big) \sum_{i\le r} \lambda_i, \end{aligned}$$

which rearranges to the stated $\Delta f(r)$. At an integer minimizer $r^\star$ of f on the grid, the step-2 discrete first-order condition is $\Delta f(r^\star{-}2) \le 0 \le \Delta f(r^\star)$ by definition of a discrete minimum. □

Solver. The rank grid $\{r_{\min}, r_{\min}{+}2, \dots, d\}$ is finite (≤ 32 candidates at $d{=}128$ with step 2), so a brute-force enumeration that sets $b = \mathrm{round}(B/r)$, clips to $[b_{\min}, b_{\max}]$, and returns $\arg\min D(r,b)$ attains the global minimum on the integer feasible set. The step-2 rank grid loses no precision in practice: changing r by ± 1 shifts $b = B/r$ by at most ≈ 0.25 bits in our budget range, and we observed the step-1 optimum coinciding with the step-2 sub-grid optimum on the heads we sampled.

C Cross-layer sum-distortion minimization (C2)

Assumption C.1 (Discrete convexity of the head-level envelope)**.** For every (layer, head), the integer sequence $B \mapsto D^\star_{\ell,h}(B)$ of C1-optimal distortions is non-increasing and discrete-convex: $D^\star_{\ell,h}(B+1) - 2D^\star_{\ell,h}(B) + D^\star_{\ell,h}(B-1) \ge 0$ for every interior integer B.

Strict discrete-convexity can locally fail at (r,b) grid transitions, so we treat Assumption C.1 as a working approximation; the empirical impact on the surrogate optimum is small (§D).

Proposition C.2 (Cross-layer sub-gradient KKT under Assumption C.1; multi-channel sum-distortion minimization [14, 49])**.** *Under Assumption C.1, let $\widetilde{D}^\star_{\ell,h} : \mathbb{R}_{\ge B_{\min}} \to \mathbb{R}_{\ge 0}$ be the piecewise-linear envelope of $D^\star_{\ell,h}$ (linear interpolation between consecutive integer grid points). $\widetilde{D}^\star_{\ell,h}$ is then convex, non-increasing, and continuous, with sub-differential*

$$\partial \widetilde{D}^\star_{\ell,h}(B) = \big[D^\star_{\ell,h}(B) - D^\star_{\ell,h}(B-1),\ D^\star_{\ell,h}(B+1) - D^\star_{\ell,h}(B)\big] \qquad \text{at integer } B$$

(left and right difference quotients, in the standard order $f'_-(B) \le f'_+(B)$). Consider the relaxation

$$\min_{\{B_{\ell,h}\} \in \mathbb{R}^{LH}_{\ge B_{\min}}} \ \sum_{\ell,h} \widetilde{D}^\star_{\ell,h}(B_{\ell,h}) \quad \text{s.t.} \quad \sum_{\ell,h} B_{\ell,h} = B_{\mathrm{tot}}. \tag{10}$$

At any stationary point there exists $\nu^\star \ge 0$ such that for all heads with $B^\star_{\ell,h} > B_{\min}$,

$$-\nu^\star \ \in\ \partial \widetilde{D}^\star_{\ell,h}(B^\star_{\ell,h}), \tag{11}$$

and $B^\star_{\ell,h} = B_{\min}$ for the remaining heads (complementary slackness). On interior points where $\widetilde{D}^\star_{\ell,h}$ is differentiable, this reduces to the classical statement that marginal distortions are equalized.

Proof. Under Assumption C.1 the piecewise-linear envelope $\widetilde{D}^\star_{\ell,h}$ is convex and continuous with the stated sub-gradients, so (10) is the minimization of a sum of closed convex functions subject to one affine equality and LH affine inequalities. Slater's condition is trivially satisfied by $B_{\ell,h} = B_{\mathrm{tot}}/(LH) \ge B_{\min}$ whenever B_{tot} is large enough [49, Section 5.2.3]. The associated Lagrangian is

$$\mathcal{L}(\{B_{\ell,h}\}; \nu, \{\rho_{\ell,h}\}) = \sum_{\ell,h} \widetilde{D}^\star_{\ell,h}(B_{\ell,h}) + \nu \cdot \Big(\sum_{\ell,h} B_{\ell,h} - B_{\mathrm{tot}}\Big) - \sum_{\ell,h} \rho_{\ell,h} \cdot (B_{\ell,h} - B_{\min}), \quad \rho_{\ell,h} \ge 0,$$

and the sub-gradient optimality conditions at a primal–dual saddle point are (a) primal feasibility, (b) $\rho_{\ell,h} \ge 0$, (c) $\rho_{\ell,h}(B^\star_{\ell,h} - B_{\min}) = 0$, and (d) $0 \in \partial \widetilde{D}^\star_{\ell,h}(B^\star_{\ell,h}) + \nu^\star - \rho_{\ell,h}$. Specializing (d) to $\rho_{\ell,h} = 0$ gives Eq. (11). □

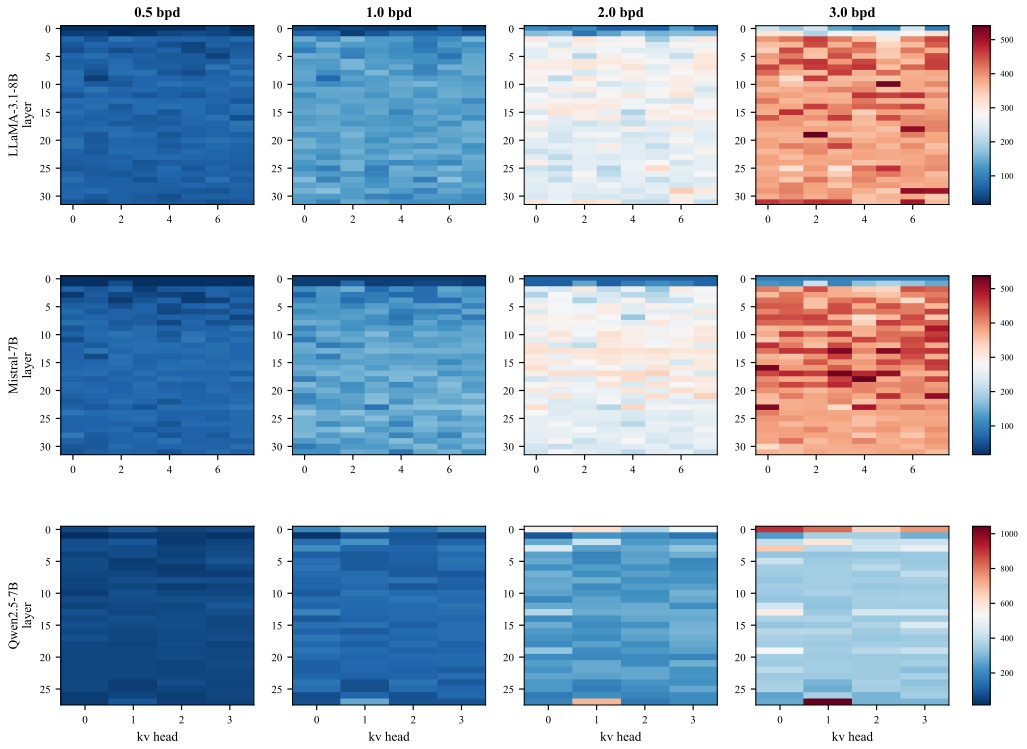


Figure A3: C2 budget allocation $B^{\star}_{\ell,h}$ on K for LLaMA-3.1-8B, Mistral-7B, Qwen2.5-7B at four target bpds. Cell color is the per-head bit budget (red high, blue low; scale normalized per panel). Non-flat pattern $\Rightarrow$ heterogeneity worth exploiting across all three models.

Damped proportional update. Eq. (11) characterizes the optimum but does not specify a solver. We use a heuristic that targets a simpler condition, value equalization $D^{\star}_{\ell,h} = \bar{D}$ with $\bar{D} = \frac{1}{LH}\sum_{\ell,h} D^{\star}_{\ell,h}$. This is a first-order proxy for the KKT condition near uniform: under $D \propto B^{-\alpha}$, marginal equalization $\partial D/\partial B = -\nu^{\star}$ becomes $\alpha D_{\ell,h}/B_{\ell,h} = \nu^{\star}$, which reduces to value equalization $D_{\ell,h} = \nu^{\star}\bar{B}/\alpha$ exactly when all $B_{\ell,h} \approx \bar{B}$. Damping the iteration keeps budgets close enough to $\bar{B}$ for this approximation to remain accurate. Under the empirical scaling $\alpha \approx 2$ (the 2^{-2b} falloff of uniform quantization), heads above the mean require a budget boost roughly proportional to $\sqrt{D^{\star}_{\ell,h}/\bar{D}}$. Damping by $\eta = 0.3$ around the mean budget $\bar{B}$ gives the update

$$B^{(t+1)}_{\ell,h} = \max\Big(B_{\min},\ B^{(t)}_{\ell,h} + \eta\,\bar{B}\big(\sqrt{D^{\star}_{\ell,h}/\bar{D}} - 1\big)\Big), \tag{12}$$

followed by total-budget renormalization $B_{\ell,h} \leftarrow B_{\ell,h} \cdot B_{\mathrm{tot}}/\sum B_{\ell,h}$. At the fixed point $D^{\star}_{\ell,h} = \bar{D}$ for all heads (zero update at equality), the sign is always correct, and the per-iteration cost is one C1 call per head ($O(LH)$). On every tested model the iteration reaches $<1\%$ per-head budget change within five rounds, which we use as the default. The resulting per-head budgets are non-flat across all three models (Figure A3), confirming that cross-head heterogeneity is worth exploiting.

D C2 solver choice: proportional update vs. exact DP

The solver used throughout the paper is the damped proportional update of (12) (§C). It is not surrogate-optimal: it compares $D^\star_{\ell,h}$ to the mean $\bar{D}$ rather than to a subgradient, and the integer constraint is enforced only by the final rounding. A natural question is whether a more careful solver would measurably improve PPL.

Alternative solver: exact DP. The integer problem

$$\min_{\{B_{\ell,h}\}\in\mathbb{Z}^{LH}} \sum_{\ell,h} D^\star_{\ell,h}(B_{\ell,h}) \quad \text{s.t.} \quad \sum_{\ell,h} B_{\ell,h} = B_{\text{tot}},\ B_{\ell,h} \in [B_{\min}, B_{\max}]$$

is a separable knapsack and admits an exact $O(LH \cdot B_{\text{tot}} \cdot C)$ dynamic-programming solver, where $C = (B_{\max} - B_{\min})/\Delta$ is the per-head candidate count at step Δ. This DP makes *no* convexity assumption on $D^\star_{\ell,h}$: it finds the surrogate-optimal integer allocation even when the head-level envelope has plateau-then-jump patterns that Assumption C.1 rules out.

PPL on real models. Table A1 reports Wikitext-2 PPL on LLaMA-3.1-8B, Mistral-7B, and Qwen2.5-7B (k-only) for our solver and DP, with C1 identical across columns.

Table A1: C2 solver sensitivity on Wikitext-2 PPL (k-only, single seed). **Ours** is the damped proportional update used in the paper. **DP-optimal** solves the integer surrogate exactly without any convexity assumption. PPL: cells where one method beats the other are bolded. Rightmost column: wall-clock ratio of the C2 step ($t_{\text{DP}}/t_{\text{ours}}$, single-CPU-thread, calibration excluded). At 1.0 bpd, our solver costs 130 ms on LLaMA / Mistral (256 heads each) and 58 ms on Qwen (112 heads); DP costs 5.3 s and 2.3 s. The ratio grows with bpd because DP's table size is $O(LH \cdot \bar{B})$ while our iteration count stays at 5.

Model	bpd	PPL ours	PPL DP-optimal	$t_{\text{DP}}/t_{\text{ours}}$
LLaMA-3.1-8B	0.5	**15.56**	17.02	25×
	1.0	**15.55**	15.71	40×
	2.0	**6.78**	7.38	49×
	3.0	5.98	**5.93**	50×
	4.0	5.80	**5.76**	51×
Mistral-7B	0.5	7.57	**7.32**	26×
	1.0	6.16	**6.08**	40×
	2.0	**5.02**	5.17	51×
	3.0	**4.84**	4.84	51×
	4.0	4.79	**4.78**	50×
Qwen2.5-7B	0.5	12.30	**11.92**	25×
	1.0	9.63	**9.28**	39×
	2.0	**6.97**	8.16	49×
	3.0	**6.35**	6.36	47×
	4.0	**6.17**	6.18	47×

Takeaway. Across the 15 cells in Table A1, the two solvers give comparable PPL: the median per-cell gap is 0.01, the mean is 0.52, and individual gaps fall within the ±0.1–0.5 multi-seed noise. Wall-clock differs much more: 25–51× across the bpd range (one-time calibration cost). With no measurable PPL gain from DP we adopt the cheaper solver as the default; the DP solver ships alongside in the reference code and can be selected with a single flag.

Where the surrogate deviates. A natural follow-up is why surrogate optimality does not translate into PPL. To find out, we quantized every (layer, head)'s rotated calibration activations with the deployed quantizer at $b = 1, \ldots, 4$ and compared the measured MSE against the $q(b) \propto 2^{-2b}$ model, normalized at b=4 so that only the shape matters (Table A2). The model is accurate at $b \geq 3$ but underestimates the true distortion by 1.7× at b=2, with a 30–40% head-to-head spread. An allocator compares distortion across heads and bit widths, so what hurts is error that differs across the options being compared. An exact solver tends to pick the options whose cost is most underestimated.

Table A2: Measured quantizer MSE divided by the 2^{-2b} model (shape only, normalized at b=4, all three models). b=1 is below the deployed $b \geq 2$ floor and is shown for the trend.

	$b=4$	$b=3$	$b=2$	$b=1$
median across heads	1.0	1.1	**1.7**	6.9
per-head spread (p95−p05)/median	31 %	33 %	**41 %**	36 %

Exact DP over measured distortion. Replacing the Bennett estimate with measured per-head distortion tables (quantizing the calibration activations once per candidate (r, b) adds a few seconds to calibration) and re-running the same exact DP removes the degradation entirely (Table A3). The exact solver then improves on the shipped allocator on all three models, and the solution becomes nearly seed-invariant. The degradation therefore comes from the error of the assumed quantizer model, not from the solver or the objective. Once the estimate is measured instead of assumed, the same objective and the same solver behave well.

Table A3: Wikitext-2 PPL at 1 bpd, k-only (mean $\pm$ SD over 3 seeds).

	LLaMA	Mistral	Qwen
FP16	5.71	4.77	6.07
water-filling + Bennett estimate (shipped)	16.33 ± 0.55	7.39 ± 0.13	8.14 ± 0.07
exact DP + Bennett estimate	17.55 ± 1.49	7.17 ± 0.19	7.98 ± 0.06
exact DP + measured estimate	$\mathbf{8.11 \pm 0.02}$	$\mathbf{5.39 \pm 0.01}$	$\mathbf{7.43 \pm 0.01}$

E Effective bpd: per-method overhead derivations

Each method's nominal bpd hides a different amount of per-token metadata. We add an overhead Δ to every cell, counting only storage paid *per (channel, token)*; fixed per-head or per-layer bases are excluded as they amortize away at T=32768. Head dimension $d = 128$, raw bits (no entropy coding, applied uniformly to all methods).

KIVI [6]. Token-axis grouping with group_size $= 32$; each (group, channel) stores an fp16 $(\min, \max)$ pair: $\Delta_{\text{KIVI}} = 32/32 = 1.0$ bpd. (the configuration used in standard reproductions); KIVI's 128-token fp16 residual contributes ≤ 0.07 bpd at T=32,768 (negligible at our sequence length).

KVQuant [7]. Per-channel quantization plus top-ρ=1% fp16 outliers, following KVQuant's own convention (excluding CSR index overhead): $\Delta_{\text{KVQuant}} = \rho \cdot 16 = 0.16$ bpd. Sink tokens add $80/T$ bpd, negligible at $T \geq 1000$.

GEAR [8]. Group-wise quantization (group_size $= 64$) plus rank-r=4 residual $Q_m P^\top$ with $Q_m \in \mathbb{R}^{T \times r}$: group meta $32/64 = 0.5$, residual row $r \cdot 16/d = 0.5$, basis P assumed shared across tokens (FlexGen-style convention; per-token P would double the residual cost). $\Delta_{\text{GEAR}} = 1.0$ bpd. (excluding the sparse outlier component, which adds further overhead).

TurboQuant [11]. Random rotation + Lloyd–Max; metadata is a per-channel scale (fixed per-head). $\Delta_{\text{TurboQuant}} = 0$.

SVDq [9], KQ-SVD [10], KV-COBRA (ours). All three store a fixed SVD basis (per-layer for SVDq, per-head otherwise) and cache only the quantized latent of r channels at b bits. The per-(channel, token) cost is exactly the nominal $r \cdot b/d$; $\Delta = 0$ for all three.

Summary. KIVI and GEAR each pay $+1.0$ bpd on top of their nominal b; KVQuant pays $+0.16$; rotation-based methods pay 0. A KIVI or GEAR cell at nominal 2 bpd sits at the same effective operating point as our 3 bpd.

F Scope: V-side compression is structurally limited

Flatter spectra. Power-law fits $\lambda_i \propto i^{-\alpha}$ give median $\alpha_V/\alpha_K \approx 0.46$ across the three models (Fig. A4, top row). The flatter V spectrum pins C1 at the ceiling $r^\star{=}d$ on essentially every head at 2 bpd (bottom row), while on K the optimum is spread far below d so C1 has room to trade rank for bit width.

C2 and KL reordering add nothing. With per-head spectra nearly identical, each head's $D^\star(B)$ curve is nearly the same, so C2 has no free energy to redistribute. Empirically, the KL reordering provides no measurable gain on V, consistent with the flat-spectrum behavior.

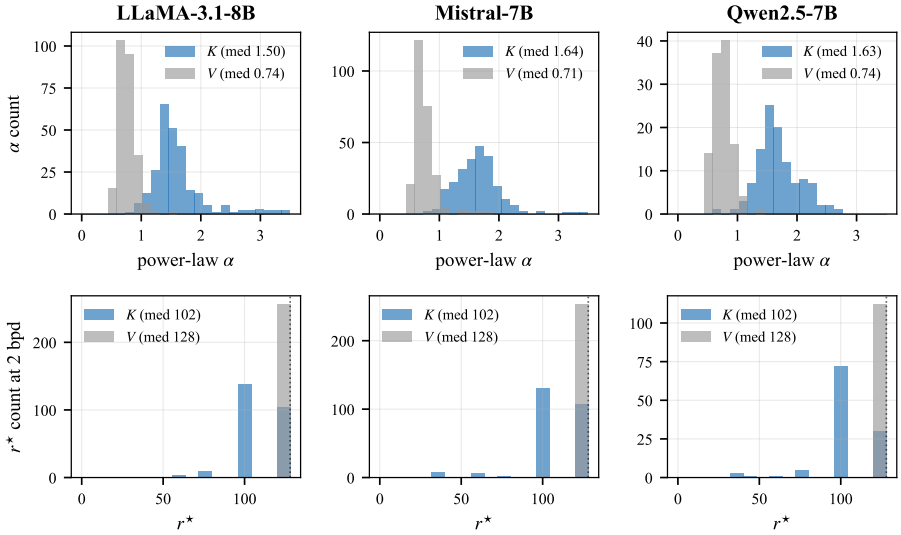


Figure A4: V is structurally flatter than K. **Top:** per-(layer, head) power-law exponents α; median $\alpha_V/\alpha_K \approx 0.46$. **Bottom:** C1-selected $r^\star$ at 2 bpd. On V, $r^\star = d{=}128$ on nearly every head (dotted line); on K the distribution is broad.

Empirical check. Figure A5 compares KV-COBRA w/o C2 (C1 only) on K-only vs. V-only at matched bit budgets. On a log scale the V-only curve sits orders of magnitude above the K-only curve through sub-3 bpd and converges to FP16 only around 4–5 bpd: V is harder to compress at low rates because truncating its flat spectrum is expensive.

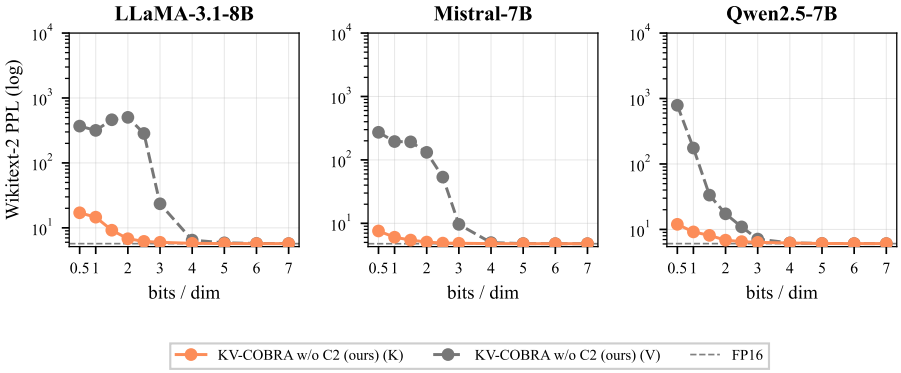


Figure A5: K-only vs. V-only Wikitext-2 PPL (log) for KV-COBRA w/o C2 on three models. Orange solid: K-only; grey dashed: V-only; dashed horizontal: FP16. V converges to FP16 only near 4–5 bpd.

What we do on V. Putting the three observations together, the V side reduces to plain bit allocation. The flat spectrum makes rank truncation lossy for any non-trivial savings, the per-head $D^\star(B)$ curves are similar enough that cross-head redistribution buys nothing (so C2 is redundant), and the KL reordering coincides with the MSE one. We therefore reuse the same kernel as on K but skip C2, and the resulting pipeline is a plain rotate-and-quantize on the V projection. The Hadamard rotation still equalizes per-channel variance and so is kept, which is consistent with the joint $K{+}V$ table where the V-side bits behave like a uniform quantizer (Tables 4, 5). Because the V side has no per-head allocation to optimize, we do not feature V-only as a primary evaluation track—the joint $K{+}V$ sweep is the meaningful test, since it lets the K side absorb the rank budget that V cannot use.

G Post-RoPE experiments

We ran every tested method under both pre-RoPE and post-RoPE compression with the same calibration protocol. Post-RoPE requires a new harness because the attention forward has to be monkey- patched to inject the compressor after the rotary embedding is applied to K. Pre-RoPE remains uniformly better (Figure A6) because the per-head covariance is position-stationary pre-RoPE and position-dependent post-RoPE: a single static SVD basis captures the structure in the first case but not the second.

The mechanism is worth spelling out. After RoPE, each position stores a differently rotated copy of the key distribution, so a static basis faces the average over positions, $\mathbb{E}_p\left[R(p)\,\Sigma\,R(p)^\top\right]$, which is much closer to isotropic than Σ itself, with the high-frequency rotation planes flattening first. The low-rank structure the method relies on fades, so any static basis pays more truncation loss at the same rank. This matches the per-channel structure mixing reported by KVQuant [7]. Position-dependent bases could track the rotation in principle, at the cost of the one-shot calibration that keeps the method simple. We leave this as future work.

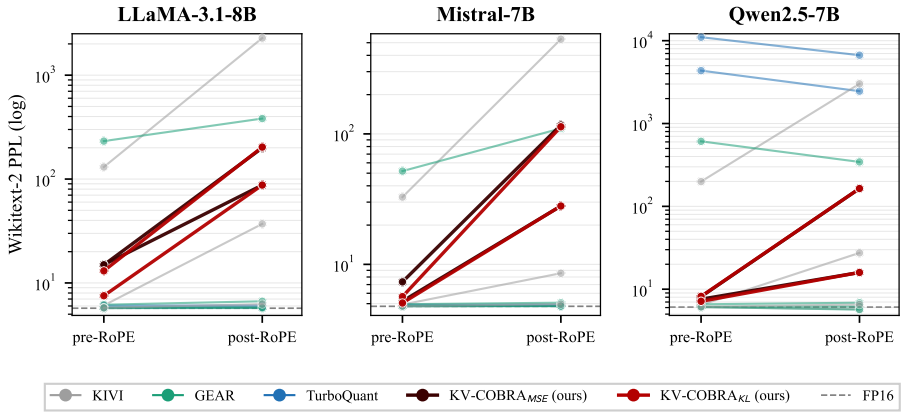


Figure A6: Pre-RoPE vs. post-RoPE PPL at the canonical bit rates, per baseline. Pre-RoPE is uniformly better: the per-head covariance is position-stationary and admits a good shared basis.

H Experimental setup in detail

This appendix expands Section 5 with the details omitted there for space.

Models. LLaMA-3.1-8B [30]; Mistral-7B-v0.3 [31]; Qwen2.5-7B-Instruct [32]. All three models use grouped-query attention and are loaded in `fp16` with their default tokenizers, providing three independently trained 7–8B backbones from different labs.

Calibration. Each calibration run passes 32 text samples of 1,024 tokens each from the Wikitext-2 [33] training split through a single forward pass with hooks on every k_{proj} and v_{proj} output. The per-head activations are centered and eigendecomposed in `float32`. Calibration is deterministic given the seed and the tokenized data; no gradient steps are taken at any point.

PPL evaluation. Perplexity is measured on Wikitext-2 [33], the Penn Treebank (PTB) [34], and the C4 English subset [35], each at $32{,}768$ evaluation tokens using a sliding window of length $2{,}048$ and stride 512. These settings match the ones used by the KIVI [6] and KVQuant [7] evaluation scripts.

LongBench and zero-shot. We report mean F1 on the LongBench [41] English subset at 50 samples per task on NarrativeQA [42], QASPER [43], MultiFieldQA-en (introduced by [41]), HotpotQA [44], and MuSiQue [45]. Zero-shot accuracy is reported on ARC-Challenge [36], HellaSwag [37], PIQA [38], WinoGrande [39], and MMLU [40] at 200 samples per task. All scores in the paper are per-task means.

Baselines and bit-width support. Every baseline reported in the paper is ported directly from the reference implementation provided by the original authors, with only enough glue code to share our `KVCompressor` dispatch. TurboQuant requires precomputed Lloyd–Max codebooks per bit-width and is restricted to $\{3, 4\}$ bpd as in the official repository. KIVI and KVQuant use uniform integer quantization with per-(group, channel) statistics that is well-defined at any integer bit-width; we evaluate them at $\{1, 2, 3, 4\}$ bpd in our sweep, while their original papers focus on $\{2, 4\}$ bpd (KIVI) and $\{2, 3, 4\}$ bpd (KVQuant).

Bit-rate sweep. Integer-only baselines run at the bit-widths supported by their official implementations (see above). KV-COBRA$_{\mathsf{MSE}}$ and KV-COBRA$_{\mathsf{KL}}$ run at $\{0.5, 1\}$ bpd in k-only mode for the multi-seed ablation (Table 3); the bit-rate scan in Fig. 4 additionally covers $\{1.5, 2, 2.5, 3, 4\}$ bpd. The $K{+}V$ joint sweep runs at the $(K_{\text{bpd}}, V_{\text{bpd}})$ splits of Tables 4 and 5. We do not run $K{+}V$ symmetric cells because the decomposition treats the two sides independently.

Seeds and variance. The pipeline (calibration, compression, evaluation) is deterministic given `torch`, `numpy`, and the Walsh–Hadamard sign seed. Single-seed runs use a fixed seed. Table 3 and Fig. 4 report mean $\pm$ 1 SD over 5 seeds across all three models. Across these multi-seed cells the median relative SD is 0.51% and the maximum is 9.06% (at 0.5 bpd).

I Additional empirical results

Per-dataset / per-task breakdown at 0.5 and 1.0 bpd. Tables A4 and A5 expand the body-level summary (Table 3) into per-dataset PPL and per-task zero-shot / LongBench breakdowns, highlighting where each objective wins. KV-COBRA$_{\mathsf{KL}}$ takes most zero-shot tasks on LLaMA, and dominates LongBench especially on Qasper, MultiFieldQA, and NarrativeQA on Mistral. KV-COBRA$_{\mathsf{MSE}}$ retains PPL advantage on LLaMA Wikitext-2 and on most Qwen metrics.

Wall-clock at the primitive level. KV-COBRA stores the cache as packed b-bit codes plus a fold-in of the Hadamard rotation into V_r, so the per-head attention primitive at decode dequantizes and applies the inverse rotation in a single fused pass. Table A6 reports the speedup of this primitive over FP16-SDPA at the same shapes ($H_q = 32$, $d = 128$), under CUDA graphs on an RTX 6000 Ada. At the retained-rank choice used in the paper ($r = 16$), KV-COBRA is 1.79–$1.89\times$ faster than FP16 across $T \in \{32\text{k}, 64\text{k}, 128\text{k}\}$; the speedup holds up to $r = 64$ and only inverts at $r = 128$ where the quantized path's BMM cost overwhelms the bandwidth savings.

Task-dependence of the K/V asymmetry. Table A7 lists the best $(K_{\text{bpd}}, V_{\text{bpd}})$ split of KV-COBRA$_{\mathsf{KL}}$ for each LongBench task at total budgets $5/6/7$ bpd, per model. The V-heavy preference holds at the level of individual tasks, not only the suite mean. The per-task best split is V-heavy in 40 of 45 (model, task, budget) cells, and all five exceptions are QASPER or MultiFieldQA, the most extraction-flavored tasks, where the best split drifts from V-heavy through balanced to K-heavy as the budget grows. The pattern matches the two error paths. K distortion perturbs *addressing* (logit noise through the softmax, most damaging when attention must be peaked), while V distortion perturbs *content* (entering the output linearly and degrading smoothly). At $V_{\text{bpd}}{=}2$ the flat V spectrum leaves no low-error directions to drop, which is why the collapse is abrupt.

Layer-position sensitivity. C1/C2 weight every head purely by the magnitude of its spectral distortion. A natural question is whether equal distortion is equally harmful at every depth. To test this we apply the full 1-bpd KV-COBRA$_{\mathsf{KL}}$ allocation to one layer quartile at a time (all other layers

Table A4: Per-dataset / per-task breakdown at 0.5 bpd (k-only). PPL (↓), zero-shot (%, ↑), LongBench F1 (↑), all aggregate 5 seeds; cells report mean ± std.

	LLaMA		Mistral		Qwen	
Metric	MSE	KL	MSE	KL	MSE	KL
Perplexity (↓)						
Wikitext-2	**16.29 ± 0.94**	17.60 ± 0.72	9.05 ± 0.31	**8.08 ± 0.07**	**10.06 ± 0.12**	10.37 ± 0.13
PTB	**30.75 ± 3.11**	34.74 ± 1.06	**81.88 ± 4.66**	86.20 ± 2.10	**19.50 ± 0.27**	20.37 ± 0.34
C4	**46.46 ± 4.10**	59.60 ± 19.14	18.57 ± 1.20	**14.44 ± 0.20**	**19.45 ± 0.16**	20.76 ± 0.30
mean	**31.17 ± 2.21**	37.31 ± 6.44	36.50 ± 1.46	**36.24 ± 0.68**	**16.33 ± 0.15**	17.17 ± 0.23
Zero-shot accuracy (%, ↑)						
ARC-C	**32.0 ± 1.4**	30.9 ± 1.0	**36.1 ± 1.2**	35.2 ± 1.1	33.0 ± 1.6	**35.5 ± 1.4**
HellaSwag	**46.9 ± 0.8**	46.6 ± 0.7	**49.7 ± 0.4**	49.7 ± 0.5	**48.2 ± 0.9**	48.1 ± 0.9
PIQA	**74.3 ± 1.4**	73.3 ± 1.4	**76.3 ± 1.1**	76.2 ± 0.6	**73.3 ± 1.7**	72.7 ± 1.9
WinoGrande	61.1 ± 3.2	**61.7 ± 0.9**	**69.0 ± 2.7**	66.9 ± 1.6	**61.6 ± 1.9**	58.2 ± 2.8
MMLU	29.5 ± 1.8	**30.2 ± 1.8**	**35.9 ± 1.0**	34.5 ± 1.0	27.5 ± 1.1	**28.1 ± 1.0**
mean	**48.8 ± 1.0**	48.5 ± 0.5	**53.4 ± 0.9**	52.5 ± 0.3	**48.7 ± 0.9**	48.5 ± 0.7
LongBench F1 (↑)						
NarrativeQA	**8.16 ± 1.31**	7.99 ± 1.16	4.25 ± 0.75	**6.12 ± 1.22**	6.99 ± 0.41	**7.36 ± 0.43**
Qasper	9.61 ± 0.54	**9.69 ± 1.27**	10.77 ± 0.51	**11.17 ± 1.25**	9.61 ± 0.46	**11.27 ± 0.91**
MultiFieldQA	**11.51 ± 1.25**	11.44 ± 0.58	14.38 ± 2.32	**18.37 ± 1.03**	15.94 ± 0.49	**16.09 ± 0.42**
HotpotQA	**3.27 ± 1.19**	2.76 ± 0.44	3.67 ± 1.40	**5.02 ± 1.40**	3.88 ± 0.83	**4.72 ± 0.74**
MuSiQue	1.61 ± 0.47	**1.88 ± 0.39**	2.02 ± 0.46	**2.98 ± 0.53**	2.01 ± 0.54	**2.06 ± 0.19**
mean	**6.83 ± 0.63**	6.75 ± 0.58	7.02 ± 0.73	**8.73 ± 0.42**	7.69 ± 0.40	**8.30 ± 0.31**

Table A5: Per-dataset / per-task breakdown at 1.0 bpd (k-only). PPL (↓), zero-shot (%, ↑), LongBench F1 (↑), all aggregate 5 seeds; cells report mean ± std.

	LLaMA		Mistral		Qwen	
Metric	MSE	KL	MSE	KL	MSE	KL
Perplexity (↓)						
Wikitext-2	16.45 ± 0.51	**13.27 ± 0.60**	7.47 ± 0.15	**5.65 ± 0.03**	**8.14 ± 0.05**	8.17 ± 0.03
PTB	25.01 ± 1.68	**21.12 ± 1.17**	59.38 ± 2.65	**32.86 ± 0.33**	**14.79 ± 0.15**	15.07 ± 0.12
C4	32.83 ± 1.27	**26.39 ± 1.08**	14.70 ± 0.61	**9.18 ± 0.07**	**15.99 ± 0.15**	16.35 ± 0.10
mean	24.76 ± 1.14	**20.26 ± 0.87**	27.18 ± 0.97	**15.90 ± 0.13**	**12.97 ± 0.09**	13.20 ± 0.06
Zero-shot accuracy (%, ↑)						
ARC-C	34.8 ± 1.2	**34.9 ± 1.2**	39.1 ± 1.6	**40.1 ± 2.0**	38.1 ± 1.2	**38.2 ± 2.0**
HellaSwag	**46.8 ± 0.7**	46.6 ± 0.8	50.1 ± 0.5	**50.9 ± 0.6**	50.8 ± 1.1	**51.6 ± 0.7**
PIQA	75.9 ± 0.7	**76.1 ± 1.4**	78.1 ± 1.3	**78.3 ± 0.8**	77.6 ± 1.2	**77.7 ± 0.9**
WinoGrande	64.6 ± 2.0	**67.2 ± 2.2**	**71.0 ± 1.5**	71.0 ± 1.7	64.3 ± 2.8	**66.2 ± 1.7**
MMLU	**35.4 ± 1.1**	33.1 ± 1.2	34.4 ± 1.2	**35.2 ± 0.5**	27.5 ± 1.3	**28.8 ± 1.7**
mean	51.5 ± 0.6	**51.6 ± 0.7**	54.6 ± 0.5	**55.1 ± 0.5**	51.7 ± 0.7	**52.5 ± 0.7**
LongBench F1 (↑)						
NarrativeQA	9.38 ± 0.40	**11.21 ± 0.39**	6.37 ± 1.22	**8.34 ± 1.16**	7.46 ± 0.36	**7.65 ± 0.34**
Qasper	9.60 ± 1.25	**11.12 ± 0.78**	11.57 ± 1.47	**12.57 ± 1.37**	12.02 ± 0.71	**13.25 ± 0.64**
MultiFieldQA	18.70 ± 1.24	**22.28 ± 1.85**	20.45 ± 1.23	**21.73 ± 1.06**	18.70 ± 0.59	**19.24 ± 0.82**
HotpotQA	5.56 ± 1.39	**8.36 ± 2.03**	5.82 ± 0.80	**7.18 ± 0.33**	4.88 ± 0.35	**5.10 ± 0.54**
MuSiQue	3.04 ± 0.41	**3.83 ± 0.43**	3.53 ± 0.32	**3.77 ± 0.59**	**2.45 ± 0.17**	2.40 ± 0.17
mean	9.25 ± 0.54	**11.36 ± 0.77**	9.55 ± 0.56	**10.72 ± 0.49**	9.10 ± 0.29	**9.53 ± 0.30**

Table A6: Per-layer attention speedup at $H_q = 32$, $d = 128$, CUDA-graph captured. Each cell is FP16-SDPA latency / KV-COBRA latency at the same T (i.e. $> 1\times$ means KV-COBRA is faster); **bold** marks $> 1\times$.

	$r = 16$		$r = 32$		$r = 64$		$r = 128$	
T	$b = 2$	$b = 4$	$b = 2$	$b = 4$	$b = 2$	$b = 4$	$b = 2$	$b = 4$
32k	**1.83×**	**1.84×**	**1.77×**	**1.73×**	**1.40×**	**1.31×**	0.44×	0.32×
64k	**1.79×**	**1.85×**	**1.82×**	**1.74×**	**1.48×**	**1.40×**	0.43×	0.36×
128k	**1.89×**	**1.88×**	**1.85×**	**1.76×**	**1.49×**	**1.38×**	0.40×	0.33×

Table A7: Best $(K_{\text{bpd}}, V_{\text{bpd}})$ split of KV-COBRA$_{\mathsf{KL}}$ per LongBench task at total budgets 5 / 6 / 7 bpd. **Bold** marks the five non-V-heavy cells.

Task	LLaMA-3.1-8B	Mistral-7B	Qwen2.5-7B
NarrativeQA	(1,4) / (1,5) / (1,6)	(1,4) / (2,4) / (3,4)	(1,4) / (2,4) / (3,4)
QASPER	(1,4) / (2,4) / (3,4)	(1,4) / **(3,3)** / **(4,3)**	(2,3) / **(3,3)** / **(4,3)**
MultiFieldQA	(1,4) / (2,4) / (1,6)	(1,4) / (2,4) / (3,4)	(2,3) / **(3,3)** / (3,4)
HotpotQA	(1,4) / (1,5) / (2,5)	(1,4) / (2,4) / (2,5)	(1,4) / (1,5) / (2,5)
MuSiQue	(2,3) / (2,4) / (2,5)	(1,4) / (2,4) / (3,4)	(2,3) / (2,4) / (3,4)

Table A8: Calibration robustness vs. the paper's n=32 WikiText-2 calibration. Allocation columns give ranges over the three models and $\text{bpd} \in \{0.5, 1, 2\}$. LongBench is the five-task F1 mean of KV-COBRA$_{\mathsf{KL}}$ at 1 bpd (50 samples per task).

	Allocation vs. paper			LongBench F1		
Calibration	budget corr.	identical $b^\star$	mean $\lvert\Delta r^\star\rvert$	LLaMA	Mistral	Qwen
(reference, n=32 WikiText-2)	—	—	—	10.8	10.6	9.4
Disjoint WikiText-2 draw	0.992–1.000	97–100 %	0.45–1.17	9.6	10.0	9.4
PTB	0.984–0.998	94–100 %	0.76–3.05	12.1	10.5	8.9
C4	0.990–0.999	94–100 %	0.42–2.30	9.0	9.8	9.4
$n = 8$ (4× less data)	0.991–0.999	91–100 %	0.30–3.31	10.6	10.1	9.2
$n = 128$ (4× more data)	0.993–0.999	96–100 %	0.37–1.79	10.3	9.8	9.4

FP16) and normalize each quartile's Wikitext-2 PPL increase by its share of the distortion surrogate at the deployed (r, b) (mean over three calibration seeds). Depth does matter. Per unit of surrogate distortion, the most sensitive quartile does 2.5–5.9× more damage than the least sensitive one, and the profile is model-dependent (LLaMA is most sensitive in the first quartile, Mistral in the early half, Qwen in the mid-to-late layers). Errors also compound across depth. Compressing all layers at once costs more than the sum of the per-quartile costs (+7.72 vs. Σ=4.03 PPL on LLaMA). The current objective captures part of the depth effect through each layer's spectrum but does not reweight across layers. Folding a depth weight measured at calibration time into C2 is a natural extension.

Calibration robustness. We recomputed the full C1+C2 solution under five alternative calibrations and compared each against the paper's n=32 WikiText-2 draw (Table A8). Per-head budgets, chosen bit widths, and downstream scores all stay in a narrow band, and changing the calibration size perturbs the allocation about as much as re-drawing a same-size set. Calibrating on Python source instead of WikiText-2 moves HumanEval pass@1 by at most 1.6 points, so the allocation also transfers across domains.

Scale and family check. Table A9 repeats the three-metric comparison at 1 bpd (k-only) on LLaMA-3.1-70B, Qwen2.5-72B, and Mixtral-8x7B (a mixture-of-experts model whose attention and KV structure match Mistral-7B, with MoE only in the feed-forward blocks). The pattern from the 7–8B models carries over. KV-COBRA$_{\mathsf{KL}}$ stays closest to FP16 on every metric in all three families, while a different integer-grid baseline collapses on perplexity in each. One-time calibration at 70B takes 18.3 s on four 48 GB GPUs (14.6 s calibration forward passes, 2.2 s eigendecomposition of all 640 KV heads, 0.4 s C1+C2 solve with Hadamard fold-in, 1.1 s KL reordering), so per-head SVD is not a bottleneck at scale.

Table A9: Scale and family check at 1 bpd (k-only). PPL (↓) is the mean over Wikitext-2/PTB/C4. Zero-shot (↑) is the mean over ARC-C, HellaSwag, PIQA, WinoGrande, and MMLU (200 samples each, with mean $\pm$ SD over 3 seeds for KV-COBRA and seed-invariant single runs otherwise). LongBench (↑) is the five-task F1 mean (25 samples per task). Shaded cell marks the best compressed method per column.

Model	Method	PPL	Zero-shot	LongBench
LLaMA-3.1-70B	FP16	5.38	60.4	14.3
	KV-COBRA$_{\mathsf{KL}}$	6.97	58.4 ± 0.6	13.5
	KV-COBRA$_{\mathsf{MSE}}$	9.50	58.5 ± 0.9	10.5
	KQ-SVD	9.79	57.0	11.2
	KVQuant	$>10^3$	55.9	5.1
	KIVI	163.8	52.6	3.7
	GEAR	245.1	51.4	5.6
Qwen2.5-72B	FP16	6.27	58.0	16.3
	KV-COBRA$_{\mathsf{KL}}$	7.62	56.7 ± 0.7	16.8
	KV-COBRA$_{\mathsf{MSE}}$	7.97	57.1 ± 0.5	14.2
	KQ-SVD	7.69	54.9	13.6
	KVQuant	910.1	54.4	4.1
	KIVI	17.8	52.2	10.0
	GEAR	50.7	50.9	8.1
Mixtral-8x7B	FP16	7.95	53.4	11.3
	KV-COBRA$_{\mathsf{KL}}$	13.19	52.3 ± 0.8	11.3
	KV-COBRA$_{\mathsf{MSE}}$	13.36	51.8 ± 0.1	10.0
	KQ-SVD	27.02	52.2	7.7
	KVQuant	323.5	49.1	3.6
	KIVI	58.0	44.7	6.6
	GEAR	96.0	43.9	7.0

J Long-context retrieval on RULER

This appendix collects the RULER [50] results, mean accuracy over the 13 subtasks at T=4096 (k-only). RULER probes retrieval directly and separates the two objectives far more than PPL does. The KL reordering's margin over MSE is largest here (Table A10), consistent with the attention-quality framing of Section 4. At 2 bpd KV-COBRA$_{\text{KL}}$ reaches 92.5/90.6/90.9 (single seed), within 0.7–3.6 points of FP16. Figure A7 sweeps the full bit-rate range.

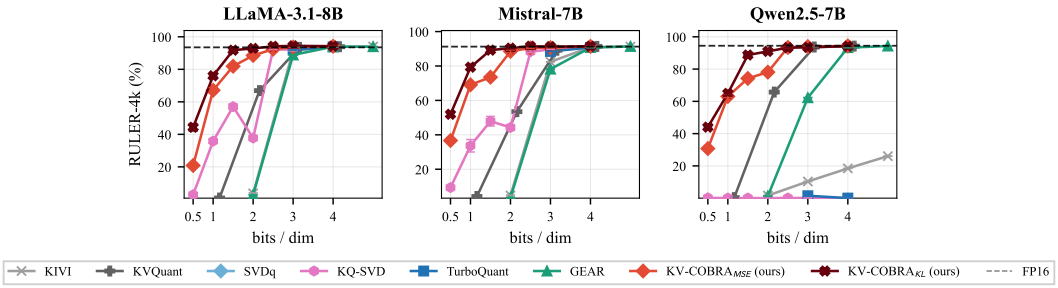


Figure A7: RULER-4k mean accuracy versus bit rate (k-only). Dashed lines mark FP16. Error bars are ± 1 SD where multi-seed data exist.

Table A10: RULER-4k mean accuracy (↑, mean over the 13 RULER subtasks at T=4096, k-only, mean $\pm$ SD over 5 seeds).

	LLaMA-3.1-8B		Mistral-7B		Qwen2.5-7B	
bpd	KV-COBRA$_{\text{MSE}}$	KV-COBRA$_{\text{KL}}$	KV-COBRA$_{\text{MSE}}$	KV-COBRA$_{\text{KL}}$	KV-COBRA$_{\text{MSE}}$	KV-COBRA$_{\text{KL}}$
0.5	20.9 $\pm$ 1.1	**44.3 $\pm$ 2.3**	36.7 $\pm$ 1.1	**52.0 $\pm$ 0.7**	30.7 $\pm$ 0.8	**44.0 $\pm$ 0.7**
1.0	67.1 $\pm$ 1.7	**76.0 $\pm$ 0.7**	69.1 $\pm$ 1.2	**79.3 $\pm$ 1.4**	63.0 $\pm$ 0.7	**64.9 $\pm$ 1.2**
FP16	93.5		91.3		94.5	

Joint K+V splits on RULER. Table A11 repeats the joint K+V sweep of Table 4 on RULER-4k. The K/V asymmetry *reverses* relative to PPL and LongBench. The best cells are balanced to K-heavy, and starving K (K_{bpd}=1) costs 15–30 points, whereas on PPL the V-heavy splits dominate. This matches the two error paths of Section 7. Retrieval needs peaked attention on a single token, so the extra bits buy the most on the addressing side.

Table A11: K+V joint compression, RULER-4k mean accuracy (↑). Same splits as Table 4. Shaded cell marks the best score within each (model, T) block. KQ-SVD's near-zero Qwen scores mirror its PPL divergence in Table 4.

		$T=5$			$T=6$				$T=7$					
	Method	(1, 4)	(2, 3)	(3, 2)	(1, 5)	(2, 4)	(3, 3)	(4, 2)	(1, 6)	(2, 5)	(3, 4)	(4, 3)	(5, 2)	FP16
LLaMA-3.1-8B	KV-COBRA$_{\text{MSE}}$	67.33	87.61	92.07	68.65	88.33	93.71	93.11	68.60	88.24	93.72	93.88	93.43	
	KV-COBRA$_{\text{KL}}$	76.53	92.35	93.60	75.86	92.77	94.02	93.53	76.80	92.82	94.31	94.24	93.50	94.11
	KQ-SVD	33.68	33.73	87.98	33.73	36.20	91.46	92.98	34.88	36.16	92.08	93.71	93.03	
Mistral-7B	KV-COBRA$_{\text{MSE}}$	67.87	86.74	90.06	67.11	87.47	90.75	90.93	67.63	87.28	90.49	91.50	90.75	
	KV-COBRA$_{\text{KL}}$	79.66	88.99	90.34	80.40	89.46	91.21	91.04	80.86	89.35	91.08	91.22	91.15	91.52
	KQ-SVD	35.58	42.12	86.37	35.86	42.86	89.33	90.04	36.34	43.47	89.99	91.53	90.42	
Qwen2.5-7B	KV-COBRA$_{\text{MSE}}$	62.32	78.01	93.18	61.79	78.05	93.89	93.85	62.76	78.26	93.94	94.22	93.92	
	KV-COBRA$_{\text{KL}}$	63.31	91.41	90.79	64.79	91.09	93.26	93.16	63.95	91.28	93.66	94.41	93.66	94.50
	KQ-SVD	0.06	0.05	0.05	0.10	0.00	0.01	0.00	0.11	0.01	0.03	0.02	1.43	

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state three contributions—(i) a joint per-head rank-bit optimizer with optimality conditions (Section 3), (ii) a query-aware attention-KL reordering (Section 4), and (iii) graceful sub-1 bpd compression (Section 6)—each supported by the corresponding experimental and theoretical results in the body.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 contains an explicit "Limitations" paragraph covering four axes: (i) scope—our comparison is restricted to per-token compressors, excluding token-axis eviction (H2O, StreamingLLM) and entropy coding; (ii) wall-clock—we report primitive-level attention speedup but not end-to-end latency; (iii) calibration stationarity—$(r^\star, b^\star)$ and budgets are computed once from 32 Wikitext-2 samples, with cross-domain transfer and refresh-during-generation not studied; and (iv) distortion model—Bennett's leading-order $q(b)$ is tight under Hadamard Gaussianization (Appendix A.3), but the finite-rate correction at $b{=}1$ can shift C1 by ± 1 direction.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms

that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Each theoretical result is paired with its assumptions and a proof. Lemma A.1 (per-channel quantization MSE) and Lemma A.2 (Eckart–Young truncation) are stated and proved in Appendix A. Lemma A.4 (head-level $D(r, b)$) is proved under Assumption A.3 in Appendix A. Proposition B.1 (per-head discrete first-order condition) is proved in Appendix B. Proposition C.2 (cross-layer sub-gradient KKT) is stated and proved under Assumption C.1 in Appendix C, with the working-approximation status of the assumption explicitly flagged. Proposition 4.1 (attention-KL upper bound) is stated in Section 4 with its three-step derivation (Taylor expansion, Fisher tight bound, query–key factorization) given inline.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 and Appendix H fully specify the experimental pipeline. Models are LLaMA-3.1-8B, Mistral-7B-v0.3, and Qwen2.5-7B-Instruct, all loaded in fp16. Calibration uses 32 Wikitext-2 samples of 1,024 tokens through a single forward pass with hooks on every k_{proj} and v_{proj} output, eigendecomposed in float32. PPL is measured on Wikitext-2/PTB/C4 at 32,768 tokens with sliding window 2,048 and stride 512; LongBench uses 50 samples per task; zero-shot uses 200 samples per task. The C1 enumeration is described in Section 3.2, the C2 update rule is given as Eq. (12), and the Hadamard rotation in Section 3.4. Every baseline is ported from its official reference implementation. The pipeline is deterministic given the seed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Anonymized reference code for the KV-COBRA method is provided as supplementary material, including the C1 enumeration solver, the C2 damped proportional update and the alternative DP solver discussed in Appendix D, the Hadamard rotation kernel, and the calibration and inference pipeline (Algorithm in Table 2). Baseline implementations (KIVI, KVQuant, GEAR, TurboQuant, SVDq, KQ-SVD) are not included; they can be reproduced from each method's official reference repository as cited in the paper. Evaluation harnesses for PPL, zero-shot, and LongBench are also not bundled, but the configurations needed to reproduce them (sliding window, sample counts, sample selection) are fully specified in Appendix H. All datasets used are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 states the experimental setup at the level of detail needed to interpret the results, and Appendix H provides the full configuration: model versions and dtype, calibration

corpus and protocol (32 Wikitext-2 samples, 1,024 tokens, fp32 eigendecomposition), evaluation windows (sliding window 2,048, stride 512, 32,768 tokens), task sample counts (50 per LongBench task, 200 per zero-shot task), per-baseline bit-width support, and the bit-rate sweep range. The C2 hyperparameters ($\eta = 0.3$, 5 rounds, $b \in [2, 8]$) are stated in Section 3.2 and Appendix C. No gradient-based training is involved; the entire pipeline is calibration-only and gradient-free, so optimizer-related hyperparameters do not apply.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Multi-seed cells in Table 3 and Figure 4 report mean $\pm$ 1 standard deviation over 5 seeds across all three models (LLaMA, Mistral, Qwen), where the seed controls torch, numpy, and the Walsh–Hadamard sign flip. The pipeline is otherwise deterministic. Per-seed standard deviations are reported per cell, and the median relative SD across multi-seed cells is 0.51% with a maximum of 9.06% (at 0.5 bpd), as stated in Appendix H. Tables A4 and A5 provide the per-task / per-dataset breakdown with the same convention. Single-seed results in Tables A1, 4, and 5 use a fixed seed and are explicitly labeled as such.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Calibration is gradient-free and runs on a single GPU; Appendix D reports C2 wall-clock at 130 ms (LLaMA / Mistral, 256 heads) and 58 ms (Qwen, 112 heads) per run on a single CPU thread, with full C1+C2 calibration completing in under 200 ms across all three models. The wall-clock measurements in Appendix I (Table A6) are obtained on an NVIDIA RTX 6000 Ada under CUDA graphs at the listed sequence lengths $T \in \{32\text{k}, 64\text{k}, 128\text{k}\}$. Evaluation runs (PPL, zero-shot, LongBench) use a single GPU with the listed sample counts; the compression pipeline

is gradient-free and adds no per-token overhead beyond the standard SVD-rotate-and-quantize kernel.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. The work studies post-training compression of attention key-value caches in publicly released large language models, uses only publicly available datasets (Wikitext-2, PTB, C4, LongBench, ARC-c, HellaSwag, PIQA, WinoGrande, MMLU), involves no human subjects, and does not introduce new pretrained models.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive impact: KV-cache compression reduces the memory bandwidth required for LLM inference, which may reduce inference energy consumption and improve the accessibility of long-context inference on memory-limited hardware. Negative impact: like other inference-efficiency techniques, more efficient LLM serving lowers the marginal cost of generating text and may indirectly facilitate large-scale generation of low-quality or misleading content; this risk is inherited from the underlying language models rather than introduced by compression itself, and is mitigated only at the model-deployment layer (which is out of scope here).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any new pretrained models, image generators, or scraped datasets. KV-COBRA is a post-training compression technique applied at inference time to existing publicly released language models; it introduces no new high-misuse-risk artifacts that would require safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets (models, datasets, baseline implementations) are cited at the point of use. The three backbone models (LLaMA-3.1-8B, Mistral-7B-v0.3, Qwen2.5-7B-Instruct) and all benchmark datasets are publicly released for academic research under their respective licenses (Llama Community License, Apache 2.0, MIT, or CC-BY). Baseline code is ported from the original authors' official repositories.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The supplementary material includes anonymized reference code for the KV-COBRA method as a new asset, with a README documenting installation, configuration, and entry points for the C1, C2, and Hadamard components. Inline comments cross-reference the corresponding equations and algorithm steps in the paper (Eq. (4), Eq. (12), and Table 2). No new datasets or pretrained model checkpoints are released.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects research. All experiments are computational evaluations on publicly available benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects research and therefore does not require IRB approval or equivalent ethical review.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as part of the core methodology, theoretical derivations, or experimental design of this research. The paper studies compression of attention key-value caches in LLMs, but the LLMs themselves are the object of study (compressed at inference time), not a methodological component contributing to the algorithmic or scientific novelty of the work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.