HYBRIDQUANT: CALIBRATION-FREE ENTROPY-CODED E_8 LATTICE QUANTIZATION OF THE KV CACHE

Anonymous authors
Paper under double-blind review

ABSTRACT

Compressing the key–value (KV) cache is central to long-context inference: for the compact and distilled models deployed where memory is scarce, the cache, not the weights, sets the limit. Established quantizers force a trade-off: calibration-free methods use fixed-rate scalar codes that leave a large gap to the rate–distortion limit, while entropy-coded compressors close the gap but fit their models to calibration data. An entropy code needs no calibration if the vectors it sees are Gaussian, isotropic and nearly independent across coordinates. HybridQuant achieves this with three rounds of the randomized fast Walsh–Hadamard transform, $3d \log_2 d$ operations per vector instead of the d^2 of a dense random rotation, then rounds to the E_8 lattice and codes with one static rANS code built for a Gaussian source; the step follows from the target rate alone, with every side bit counted. We prove how much rotation is needed: one round leaves a spike far from Gaussian (Kolmogorov distance 0.34), two bring every input close but leave a floor of order $d^{-1/2}$, and three lower it to order $1/d$ up to a logarithm (0.035 versus 0.0097 at $d = 128$, computed exactly). A dithered variant has input-independent errors, unbiased attention logits and an attention concentration bound that does not grow with context length, and an exact, calibration-free centering removes the collapse that the projection biases of Qwen2 models cause in rotation-based quantizers. On Qwen2.5-7B, Mistral-7B-v0.3 and Llama-3.1-8B the rate is within 0.003 bits of its target, and at 2.125 bits TurboQuant-MSE has 8.8%, 13.3% and 31.3% higher perplexity than HybridQuant, and a rotated entropy-coded scalar quantizer 2.3%, 3.0% and 6.5% higher, all significant under a paired bootstrap.

1 INTRODUCTION

During decoding, a transformer reads its whole key–value (KV) cache at every step, so the cache sets the memory footprint and much of the latency of long-context inference, and quantizing it to 2–3 bits per element is a standard goal (Liu et al., 2024b; Hooper et al., 2024; Zandieh et al., 2026). For compact and distilled models, deployed where memory is scarce, the cache soon outweighs the model: at 128k tokens the fp16 cache of Llama-3.1-8B takes 16 GiB, as much as its weights. We target a setting that matters for deployment: each key and value vector is quantized on its own when it enters the cache, and the quantizer must (i) need no calibration data or per-model fitting, (ii) meet a rate fixed in advance, with every side bit counted, and (iii) work on every model family. Such a quantizer is a drop-in component whose memory budget is known before the model is loaded.

The dilemma. Rotation-based scalar quantizers (Ashkboos et al., 2024; Zandieh et al., 2026; Han et al., 2026) satisfy (i) and (ii): a random rotation makes the coordinates of every vector look Gaussian, so one fixed codebook serves all of them. But a fixed-rate scalar code is inefficient at low rate. For a Gaussian coordinate at 2 bits the best 4-level quantizer reaches an SNR of 9.30 dB, 2.7 dB below the rate–distortion bound of 12.04 dB (Gray & Neuhoff, 1998). Entropy coding closes almost half of this gap (at high resolution, entropy-coded uniform quantization is 1.53 dB from the bound), and a good lattice closes more: E_8 has a normalized second moment 0.65 dB below that of the integer grid (Conway & Sloane, 1999). Entropy coding, however, needs a probability model, and established entropy-coded KV compressors fit it to data: KVTC (Staniszewski & Łańcucki, 2026) learns a PCA basis and a bit allocation from calibration data, and CacheGen (Liu et al., 2024a) pro-

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

Table 1: Claims, evidence and scope. P: proved; S: proof sketch; M: measured.

Claim	Evidence	Scope and status
The rate is fixed in advance, with all side bits counted	measured rate within 0.003 bits of target in every run with $R \geq 1$ (Table 2, Appendix B.2)	three models, all rates (M)
Rounds make one static code model valid for every input	Theorem 4.1; synthetic outliers (Table 14)	spike analyzed (not proved worst); $R = 3$ upper bound (S), rest (P); on real K/V one round suffices (M)
The third round costs little	Table 23	unfused kernels (M)
HybridQuant beats fixed-rate quantizers at equal rate	Table 2 and Figure 2, paired bootstrap	three models, two datasets (M)
E_8 and rANS each help at matched configuration	Table 5	Qwen2.5-7B (M)
HybridQuant-G: input-independent error; attention bound independent of T	Theorems 4.2 and 4.3	dither independent per position, $T \leq 40960$, $\alpha = 1$ (P)
Centering removes the Qwen2 collapse, exactly and for free	Proposition L.1; Tables 2 and 5	any quantizer (P, M)

files symbol distributions offline for each model. This breaks (i), and since the length of an entropy code depends on how well its model fits the data, it also makes the rate only as predictable as the calibration set is representative.

Gaussianity is the key. One entropy code, designed offline for a Gaussian source, is valid for every model if the vectors it codes are Gaussian, isotropic and nearly independent across coordinates; its step then follows from the target rate alone. HybridQuant obtains this with three rounds of the randomized fast Walsh–Hadamard transform (FWHT), $3d \log_2 d$ additions per vector instead of the d^2 of a dense rotation, rounds to E_8, and codes the lattice points with one static rANS model (Duda, 2013); nothing is fitted to the model. The idea stands or falls with the rotation, so we ask how much is needed in the worst case, including the massive activations of real models (Sun et al., 2024). One round is not enough; two bring every input close to Gaussian but leave a floor of order $d^{-1/2}$ on a spike; three lower it to order $1/d$ up to a logarithm (Section 4). On the models we test one round already suffices, and the third, insurance for inputs we have not seen, costs about 6% of the quantize–dequantize time.

Guarantees and a failure mode. Adding subtractive dither gives HybridQuant-G, whose error is provably independent of the input and whose attention output concentrates with constants that do not depend on the context length. Requirement (iii) exposed a failure mode: the large key and value projection biases of Qwen2 models (Liu et al., 2025a) make every rotation-based quantizer with per-vector scales that we tried collapse, TurboQuant included. We remove it with an exact, calibration-free, RoPE-correct centering that costs no bits.

Contributions.

- **Method** (Section 3): calibration-free entropy coding of the KV cache with a rate fixed in advance, in two modes, HybridQuant (default) and the dithered HybridQuant-G.
- **Theory** (Section 4): what one, two and three Hadamard rounds buy and what they provably cannot do (Theorem 4.1), an input-independent error for HybridQuant-G via the Zamir–Feder theorem (Theorem 4.2), and an attention bound that does not grow with the context length (Theorem 4.3).
- **Projection-bias centering** (Section 3.2): an exact fix for the collapse of rotation-based quantizers on models with key/value biases, which we also give to the baselines.
- **Evaluation** (Section 5): three model families, perplexity at six rates, needle retrieval to 32k tokens, LongBench-E, the published protocol of HyperQuant and OCTOPUS, and matched ablations of every component, with paired significance tests. Table 1 maps each claim to its evidence.

2 RELATED WORK

KV-cache quantization. KIVI (Liu et al., 2024b) quantizes keys per channel and values per token with asymmetric scalar codes and keeps recent tokens in full precision. KVQuant (Hooper et al., 2024) quantizes keys per channel before RoPE with non-uniform datatypes and stores outliers separately, using calibration data. QJL (Zandieh et al., 2025) stores the signs of a random projection of each key. Rotation-based methods spread outliers before scalar quantization: QuaRot (Ashkboos et al., 2024) with Hadamard transforms, TurboQuant (Zandieh et al., 2026) with a random rotation and a codebook matched to the rotated coordinates, PolarQuant (Han et al., 2026) by quantizing polar angles after a random rotation, and KVLinC (Saxena & Roy, 2025) with Hadamard rotations plus trained correction adapters. Vector quantizers with learned codebooks, CQ (Zhang et al., 2024) and CommVQ (Li et al., 2025), need calibration or training data. All of these spend a fixed number of bits per coordinate or per group. **Entropy-coded KV compression.** KVTC (Staniszewski & Łańcucki, 2026) combines a PCA basis and a bit allocation fitted on calibration data with DEFLATE coding; CacheGen (Liu et al., 2024a) arithmetic-codes quantized deltas with distributions profiled offline per model. HybridQuant is entropy-coded but calibration-free. **Lattices.** Subtractive dither makes the lattice error uniform and independent of the input (Zamir & Feder, 1992; 1996). Ordentlich & Polyanskiy (2026) analyze optimal quantization for matrix multiplication, and NestQuant (Savkin et al., 2025) uses fixed-rate nested lattices based on E_8 for weights, activations and the KV cache at 4 bits. QuIP# (Tseng et al., 2024) uses randomized Hadamard incoherence and an E_8-based codebook for weights. Two recent preprints, not peer-reviewed, are close to ours: HyperQuant (Domb et al., 2026) combines Hadamard transforms, low-dimensional lattices, Rice coding and subtractive dither, and NexusQuant (Marques, 2026) is an open-source E_8 KV quantizer; Appendix L compares both with our method. **Projection biases.** Liu et al. (2025a) report large outliers from the key and value projection biases of Qwen models, which quantizing before the bias avoids for KVQuant but not for the rotation-based QuaRot, and SageAttention (Zhang et al., 2025) observes that keys are dominated by a component shared across tokens and subtracts the token mean of K. Our centering subtracts the known weight instead, after RoPE, which makes it exact and data-independent (Section 3.2).

3 METHOD

The design follows from one requirement: the code must be fixed before any data is seen. We therefore make the data match a fixed distribution (the rotation), quantize it with the best known lattice quantizer in dimension 8 (E_8), code it with a model built for that distribution (static rANS), and derive the step size from the target rate alone. Each key vector (after RoPE) and value vector of a head, of dimension d (a power of two and a multiple of 8; $d = 128$ here), is quantized on its own when it enters the cache, with separate parameters for every layer, keys and values (Figure 1; Algorithm 1 in Appendix B).

Rotation: make every vector look Gaussian. $y = HD_3HD_2HD_1x$, where H is the normalized Sylvester–Hadamard matrix and D_1, D_2, D_3 are random sign diagonals drawn once per layer and role from a seed. Each round costs $d \log_2 d$ additions and spreads any concentrated energy over all coordinates. The rotation acts within a head, because mixing heads would change the per-head logits (Proposition E.2).

Scale and lattice: a better cell than the cube. The row RMS σ is stored in fp16 and sets the step $\Delta = s\sigma$. Each block of 8 coordinates of y/Δ is mapped to the nearest point of E_8 with the decoder of Conway & Sloane (1982a). Its normalized second moment is $G_{E_8} = 929/12960$, against $1/12$ for scalar rounding.

Entropy coding: one code for all models. Lattice points are coded with a static rANS coder with seven contexts per block that condition on earlier symbols of the same block, in independent streams of 4096 elements (Appendix B.2). The probability tables are built once, by Monte Carlo on a Gaussian source, the law that the rotated coordinates approximate (Section 4). Values the Gaussian tables never produced are coded as escapes. Decoding reverses the steps: $\hat{x} = D_1HD_2HD_3H(\Delta\hat{q})$, and attention then runs unchanged on the reconstructed keys and values.

Rate targeting and accounting: a budget known in advance. For a target of b effective bits per element, s is chosen so that the Monte-Carlo code rate equals b minus the side information, which is 16 bits of fp16 RMS per row and a 32-bit coder flush per stream ($16/d + 32/4096 = 0.133$ bits

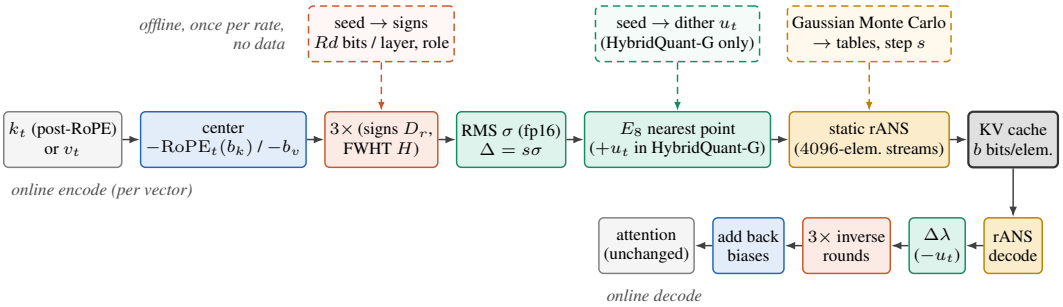


Figure 1: HybridQuant pipeline, colored by stage: centering (blue), rotation (orange), scaling and lattice (green), entropy coding (yellow); each decoding step has the color of the encoding step it inverts, and dashed boxes are offline. Offline work depends only on the target rate, not on the model or its data. Online, each key or value vector is centered, rotated by three randomized Hadamard rounds, scaled by its fp16 RMS, rounded to E_8 and entropy-coded; decoding reverses the steps with the same signs, tables and (in HybridQuant-G) dither, and attention runs unchanged.

per element at $d = 128$). The search is a bisection on the Gaussian model and uses no model data. Every reported rate is the code length of the actual lattice points under the static model, including escapes, plus all side bits; real rANS bitstreams confirm this accounting (Section 3.3).

3.1 Two operating modes

HybridQuant (the default) uses no dither. HybridQuant-G adds subtractive dither uniform on the E_8 Voronoi cell, drawn independently for every position from a seeded table of 40960 rows, longer than any sequence we evaluate. The decoder regenerates the dither from the seed, so it costs no bits. HybridQuant-G satisfies the hypotheses of Theorems 4.2 and 4.3 exactly; its cost is 1.1% perplexity at 2.125 bits and 0.1% at 3.125 bits (Table 5). Both modes use no Wiener (MMSE) scaling and no key shaping, which an ablation of our initial design showed to cost perplexity at low rate (Section 5.3, Appendix B.1).

3.2 Projection-bias centering

Qwen2 computes $k_t = W_k x_t + b_k$ before RoPE and $v_t = W_v x_t + b_v$. The biases are weights, known exactly, but they can dominate every row's norm, so a quantizer with a per-row scale spends its error budget on them and swamps the token-specific part. We quantize $\tilde{k}_t = \mathrm{RoPE}_t(k_t) - \mathrm{RoPE}_t(b_k)$ and $\tilde{v}_t = v_t - b_v$ and add the same terms back after dequantization, computing $\mathrm{RoPE}_t(b_k)$ with the model's own $\cos / \sin$ tables. Because RoPE is linear, the reconstruction error is exactly the quantizer's error on $\tilde{k}_t, \tilde{v}_t$, the model is unchanged when the quantizer is lossless, and no bits are spent (Proposition L.1). Centering does not remove the positional logit term $q \cdot \mathrm{RoPE}_t(b_k)$; it keeps that term out of the quantizer and restores it exactly. Unlike subtracting a token mean (Zhang et al., 2025), it needs no statistics of the input. We apply it to our method and to TurboQuant (marked ^{c}); models without biases, such as Mistral, are unaffected.

3.3 Complexity, storage and decoding

Figure 1 separates offline from online work. Offline, once per target rate and independent of the model, the code tables and the step are computed from a Gaussian source in a few seconds; the sign diagonals and the dither are regenerated from a seed. Online, encoding a vector costs $R(d \log_2 d + 2d)$ operations for the rounds and $O(d)$ for scaling, lattice decoding and rANS, with $O(d)$ working memory and nothing that depends on the context length. The Hadamard matrix is never stored: its only state is Rd sign bits per layer and role (48 bytes), against 64 KB for a dense rotation, and all rounds run in one kernel with one read and one write per vector. A deployment can keep the cache in the rotated domain and rotate the query instead, which replaces one inverse transform per cached vector by one per query (Appendix N).

Each (layer, KV head, role) holds a sequence of independent rANS streams of 4096 elements (32 rows of 128), so streams can be decoded in parallel and a cache is extended stream by stream. Rates differ across heads because the code is variable-length, so an implementation needs an offset per stream; a 32-bit offset would add 0.008 bits per element, which our accounting does not include. Our experiments compute static-model code lengths on the GPU. Real rANS encoding and decoding, run on the CPU on one window per model, reproduces every code bit-exactly, and the bitstreams are within 0.05% of the accounted size (always smaller) on Qwen and Mistral (Appendix B.2).

4 THEORY

The design rests on three questions, which the three results below answer: how close to Gaussian do R rounds bring an arbitrary input, what can be guaranteed about the quantization error, and does that error stay controlled as the context grows? Let $y^{(R)}$ be the output of R rounds, scaled so that each coordinate has unit variance, Φ the standard normal distribution function and d_{K} the Kolmogorov distance. Proofs are in Appendices E to I, and Table 28 lists the status of every claim. We analyse a single spike $x = \sqrt{d}\, e_m$, the limit of a massive activation and the input that maximizes the one-round Berry–Esseen bound.

Theorem 4.1 (What each round buys; Theorems E.7 and G.2 and Proposition G.1)**.** *Laws are over the random sign diagonals; (a) and (c) concern a spike.*

(a) One round depends on the input. *For a spike, every coordinate of $y^{(1)}$ is ± 1, so $d_{\mathrm{K}} = \Phi(1) - \frac{1}{2} = 0.3413$ for every d.*

(b) Two rounds are universal. *For every $R \geq 2$, every input and any $D_1, \ldots, D_{R-2}$, the law of $y_i^{(R)}$ over D_R is within $d_{\mathrm{K}} \leq 0.56\sqrt{2\log(2d/\delta)/d}$ of Φ with probability at least $1 - \delta$ over D_{R-1}, and within $0.894\, d^{-1/2}$ on average over D_{R-1}.*

(c) Two rounds leave a floor on spikes; three lower it. *For a spike, $y_i^{(2)} = (2B - d)/\sqrt{d}$ with $B \sim \mathrm{Bin}(d, \frac{1}{2})$, so $d_{\mathrm{K}} \geq \frac{1}{2}\binom{d}{d/2} 2^{-d} \geq 1/(2\sqrt{2d})$. After three rounds, $3/(8(d+1)) \leq d_{\mathrm{K}} \leq C \log d/d$ for $d \geq d_0$. The lower bound is proved; the upper bound is a proof sketch whose two lemmas rest on the local limit theorem, with constants certified numerically for $d \leq 512$. At $d = 128$, exact computation gives 0.0352 after two rounds and 0.0097 after three (Table 16).*

What this means for the design. The static code and the step rule are built for a Gaussian source, so Theorem 4.1 measures how well the design fits a spiky input after R rounds: one round cannot be trusted on such inputs, and three are the fewest that lower the two-round floor. On synthetic outliers without dither, the SNR is 16.2, 14.4, 11.9 and 11.7 dB for $R = 0, \ldots, 3$ against a design of 11.72 dB (Table 14), and on real keys and values the measured rate matches the target for every $R \geq 1$ but not for $R = 0$ (Section 5.3). Rounds also have limits, which we prove. For sparse outliers one round already gives the smallest possible peak $\|y\|_\infty$, and further rounds do not reduce it (Theorem F.2); generic-chaining bounds cannot distinguish $R = 1$ from $R = 3$, because the canonical pseudometric is the same (Proposition F.3). The case for the third round is therefore the *shape* of the distribution on spiky inputs, not its peak, and not a perplexity gain on typical inputs.

Theorem 4.2 (HybridQuant-G: the error does not depend on the input; Theorem H.2)**.** *With subtractive Voronoi dither, $\alpha = 1$ and any number of rounds, every row satisfies $\hat{y} - y = \Delta e$ with e uniform on $\mathcal{V}_{E_8}^{d/8}$ and independent of the input. Hence the per-row SNR equals the design value $1/(G_{E_8} s^2)$ for every input, and the logit error $q \cdot (\hat{k} - k)/\sqrt{d}$ has variance $G_{E_8} \Delta^2 \|q\|^2/d$.*

This is a direct application of the theorem of Zamir & Feder (1992). Our part is to make it hold exactly for an entropy-coded lattice: a row is never clipped (Lemma H.1), so large coordinates cost only rate.

Theorem 4.3 (HybridQuant-G: attention concentrates independently of context length; Theorem I.4)**.** *Fix a query q and T cached positions whose dithers are independent across positions and blocks. Let $\bar{\Delta} = \max_t \max(\Delta_t^K, \Delta_t^V)$ be the largest step in the context and $V_{\max} = \max_t \|\tilde{v}_t\| + \sqrt{d/8}\,\bar{\Delta}$, with $\tilde{v}_t$ the centered values. For every unit vector w, the attention output $\hat{o}$ satisfies $\mathrm{Var}\langle w, \hat{o}\rangle \leq (4\bar{\Delta}^2/\pi^2) L^2$ with $L^2 = 1 + 4V_{\max}^2 \|q\|^2/d$, and $\mathbb{P}\big(|\langle w, \hat{o} - \mathbb{E}\hat{o}\rangle| \geq \epsilon\big) \leq C \exp\big(-c\,\pi\epsilon/(2\bar{\Delta}L)\big)$ for universal constants c, C. None of these depends on T.*

The proof combines an ℓ_∞ Lipschitz bound for softmax, which removes T, with a Poincaré inequality for uniform laws on convex bodies. Talagrand's convex-distance inequality does not apply (softmax is not convex), and bounded differences would give a bound linear in T. The deterministic bias $\mathbb{E}\hat{o} - o$ is bounded by the same T-independent quantity (Proposition I.3), and the value bias b_v shifts $\hat{o}$ exactly (Remark I.7).

5 EXPERIMENTS

We ask four questions. Does HybridQuant beat fixed-rate quantizers at equal rate (Section 5.1)? Does the gain come from entropy coding, from E_8, or both, and what do the guarantees of HybridQuant-G cost (Sections 5.1 and 5.3)? Does it carry over to retrieval and long-context tasks? And what do the rotation rounds do on real activations (Section 5.3)?

Setup. We measure perplexity on Qwen2.5-7B, Mistral-7B-v0.3 and Llama-3.1-8B, with 64 windows of 4096 tokens from wikitext2 and C4 each, and needle retrieval and LongBench-E on Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct (Appendix T). Every key and value, in the prompt and in generated tokens, passes through the quantizer before any query uses it; keys are quantized after RoPE. Effective bits count all side information: fp16 scale and zero point per group for KIVI, the fp16 norm for TurboQuant, and the fp16 RMS and coder flushes for ours. **Baselines.** All baselines are our implementations, run in the same harness (Appendix Q.1). *TurboQuant-MSE* uses a randomized Hadamard rotation, a Gaussian Lloyd–Max codebook and an fp16 norm, at a uniform rate; it omits the two-tier outlier split used in the KV experiments of Zandieh et al. (2026) and the QJL residual. *KIVI* uses group size $g = 32$ (3.0 bits at 2-bit codes) or $g = 128$ (2.25 bits); unlike the original, it keeps no 128-token full-precision window, and in teacher-forced perplexity a key's group statistics include up to $g - 1$ later tokens, a mild advantage to KIVI. *Rot-ECSQ* is our own pipeline with the integer lattice $\mathbb{Z}$ and one round: a strong entropy-coded scalar baseline. We do not compare with KVTC or CacheGen, which need calibration or profiling (Section 2). **Statistics.** Perplexity differences are tested with a paired bootstrap over the 64 windows (10,000 resamples), and we report the relative difference $\mathrm{PPL}_B/\mathrm{PPL}_A - 1$ to a reference A with its 95% interval; retrieval accuracies have Wilson intervals, and paired retrieval comparisons use an exact McNemar test (Appendix Q). Rotations, dithers and the Monte-Carlo tables depend on a seed; over five seeds on Qwen2.5-7B wikitext2, HybridQuant has perplexity 6.941 ± 0.014 at 2.125 bits and 6.513 ± 0.006 at 3.125 bits, and HybridQuant-G 7.001 ± 0.003 and 6.513 ± 0.004 (mean $\pm$ standard deviation), well below the differences to the baselines. All runs are deterministic and checked for bit-exact reproducibility.

5.1 PERPLEXITY

HybridQuant is the best method at every rate on all three models (Table 2 and Figure 2; relative differences on wikitext2 for Qwen, Mistral and Llama, with 95% paired-bootstrap intervals). (i) *Against fixed-rate quantizers at equal rate* it is better against TurboQuant-MSE^c by 8.8% [8.2, 9.4], 13.3% [12.8, 13.8] and 31.3% [29.8, 33.0] at 2.125 bits, 4.2%, 3.5% and 7.0% at K3V2, 1.8%, 1.9% and 3.6% at 3.125 bits, and 0.5%, 0.4% and 0.9% at 4.125 bits. At 3.0 bits it is 10.1%, 6.3% and 17.8% better than KIVI-2, and at 5.0 bits it is within 0.3% of fp16. (ii) *Against our rotated entropy-coded scalar baseline*, which shares the rotation, rate targeting, rANS and centering, HybridQuant is 2.3% [2.0, 2.7], 3.0% [2.7, 3.2] and 6.5% [5.9, 7.1] better at 2.125 bits and 0.3–0.5% better at 3.125 bits. Most of the gain over fixed-rate quantizers therefore comes from entropy coding, and E_8 adds a further, significant gain that shrinks with rate, as its granular advantage predicts. (iii) *The guarantees of HybridQuant-G* cost 1.1% (Qwen) to 3.5% (Llama) at 2.125 bits and at most 0.1% at 3.125 bits. (iv) *Without centering*, TurboQuant-MSE exceeds a perplexity of 4000 on Qwen at every rate (11093, 11472, 11639 and 4207 at 2.125, K3V2, 3.125 and 4.125 bits; Table 9), and so does our method (Table 5). Domb et al. (2026) and Boss et al. (2026) keep the keys of the two outer blocks of Qwen2.5-7B-Instruct-1M in bf16 because per-vector codecs otherwise diverge. Under their protocol our quantizer diverges too without centering (perplexity 4303 at 4.125 bits); with centering, no protected layers and no full-precision window, wikitext2 perplexity rises by 10.7% at 2.125 bits (7.5× compression), against 42.0% for HyperQuant without a window (6.6×), 34.7% for OCTOPUS (2.9×) and 63.0% for TurboQuant (3.0×) as reported at 2 nominal bits (Appendix T.2).

Table 2: Perplexity (C4 / wikitext2, 64×4096 tokens). Bits: measured effective bits per element, including all side information. ^{c}: with projection-bias centering; on Mistral and Llama, which have no biases, it changes nothing. Without centering, TurboQuant-MSE exceeds 4000 on Qwen at every rate (Table 9). Bold: best in its rate group and significantly better than every other entry there (paired bootstrap, $p < 0.05$, both datasets).

		Qwen2.5-7B		Mistral-7B-v0.3		Llama-3.1-8B	
Method	Bits	C4	wikitext2	C4	wikitext2	C4	wikitext2
fp16	16	11.322	6.405	8.244	4.996	8.831	5.828
TurboQuant-MSE^c	2.125	13.066	7.529	10.173	6.204	16.468	9.679
KIVI-2 (g=128)	2.250	17.264	10.325	10.595	6.511	28.519	14.601
Rot-ECSQ ($\mathbb{Z}$, R=1)	2.123	12.366	7.081	9.244	5.638	13.095	7.854
HybridQuant-G	2.125	12.275	6.998	9.114	5.535	12.554	7.631
HybridQuant	2.124	**12.158**	**6.921**	**9.002**	**5.475**	**12.014**	**7.372**
TurboQuant-MSE^c (K3V2)	2.625	12.024	6.897	8.659	5.287	10.376	6.619
HybridQuant (K3V2)	2.624	**11.645**	**6.616**	**8.433**	**5.110**	**9.609**	**6.189**
KIVI-2 (g=32)	3.000	12.484	7.199	8.867	5.407	11.542	7.178
HybridQuant	2.999	**11.506**	**6.540**	**8.381**	**5.085**	**9.416**	**6.092**
TurboQuant-MSE^c	3.125	11.621	6.626	8.480	5.162	9.781	6.277
KIVI-3 (g=128)	3.250	11.657	6.629	8.440	5.117	9.640	6.217
Rot-ECSQ ($\mathbb{Z}$, R=1)	3.124	11.514	6.532	8.386	5.083	9.418	6.090
HybridQuant-G	3.126	11.481	6.519	8.370	5.066	9.315	6.056
HybridQuant	3.124	11.479	**6.509**	8.369	5.068	9.309	6.058
TurboQuant-MSE^c	4.125	11.401	6.466	8.297	5.028	9.039	5.934
HybridQuant	4.124	**11.357**	**6.432**	**8.269**	**5.010**	**8.940**	**5.880**
KIVI-4 (g=32)	5.000	11.354	6.424	8.261	5.008	8.891	5.861
HybridQuant	4.999	**11.332**	**6.413**	**8.252**	**5.001**	**8.862**	**5.843**

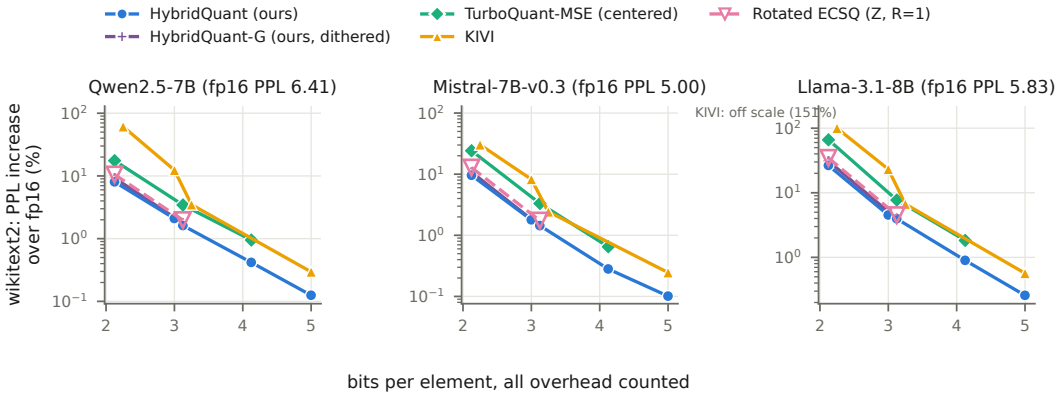


Figure 2: Perplexity increase over fp16 on wikitext2 (log scale) against storage cost; C4 is in Figure 8. The horizontal axis counts every stored bit per key or value element: the quantized values plus their overhead, that is, per-vector scales and norms, KIVI's per-group scale and zero point, and our coder's stream headers. Lower and further left is better. TurboQuant-MSE is shown with projection-bias centering; without it, it exceeds 4000 perplexity on Qwen at every rate.

5.2 Retrieval and long-context tasks

The perplexity gains carry over to retrieval. At 2.125 bits HybridQuant retrieves 90% of needles and HybridQuant-G 92%, against 75% for TurboQuant-MSE^c at the same rate (exact McNemar $p = 0.019$ and $p = 0.008$) and 84% for KIVI-2 at 3 bits ($p = 0.38$ for HybridQuant, not significant); both remain below fp16 (99%). At 3.125 bits both modes retrieve all 88 needles. Without centering, TurboQuant-MSE retrieves nothing: the large, constant key bias of Qwen2 dominates each vector's

Table 3: Needle-in-a-haystack accuracy on Qwen2.5-7B-Instruct: 11 depths × 2 trials per context length, 88 trials per method; 95% Wilson interval of the mean. Every key and value, including generated ones, is quantized.

Method	Bits	4k	8k	16k	32k	Mean [95% CI]
fp16	16	0.95	1.00	1.00	1.00	0.99 [0.94, 1.00]
KIVI-2 (g=32)	3.0	0.95	0.91	0.86	0.64	0.84 [0.75, 0.90]
TurboQuant-MSE, no centering	2.125	0.00	0.00	0.00	0.00	0.00 [0.00, 0.04]
TurboQuant-MSE^c	2.125	0.86	0.73	0.68	0.73	0.75 [0.65, 0.83]
HybridQuant-G	2.125	0.95	0.91	0.86	0.95	0.92 [0.84, 0.96]
HybridQuant	2.125	0.86	0.86	0.95	0.91	0.90 [0.82, 0.95]
v1 (initial configuration)	2.125	0.95	0.82	0.91	0.77	0.86 [0.78, 0.92]
TurboQuant-MSE, no centering	3.125	0.00	0.00	0.00	0.00	0.00 [0.00, 0.04]
TurboQuant-MSE^c	3.125	0.95	1.00	1.00	1.00	0.99 [0.94, 1.00]
HybridQuant-G	3.125	1.00	1.00	1.00	1.00	1.00 [0.96, 1.00]
HybridQuant	3.125	1.00	1.00	1.00	1.00	1.00 [0.96, 1.00]
v1 (initial configuration)	3.125	1.00	1.00	0.95	1.00	0.99 [0.94, 1.00]

Table 4: LongBench-E on Qwen2.5-7B-Instruct by category (official metrics, $\times 100$; 40 samples per task). Bits are measured; KIVI keeps generated keys in fp16.

Method	Bits	SingleQA	MultiQA	Summ.	Few-shot	Synthetic	Code	Average
fp16	16	51.8	57.1	27.5	67.7	53.8	61.0	54.3
KIVI-2 (g=32)	3.079	40.2	44.2	26.3	59.2	42.5	48.8	44.8
TurboQuant-MSE^c	2.125	29.2	44.2	25.5	61.0	40.0	45.3	42.4
v1 (initial configuration)	2.125	37.0	38.4	26.0	63.5	52.5	53.0	46.5
TurboQuant-MSE^c	3.125	48.3	46.3	27.2	64.9	51.2	60.2	50.9
v1 (initial configuration)	3.126	50.4	52.5	27.6	67.2	52.5	59.2	52.8

scale, so the quantization error swamps the token-specific part of the keys (Appendix L.1). On Llama-3.1-8B-Instruct, HybridQuant retrieves 89% against 61% for TurboQuant-MSE at 2.125 bits ($p < 10^{-4}$), and on LongBench-E it averages 49.9 against 40.0 (20 samples per task; fp16 55.3; Appendix T.1).

LongBench-E. We follow the official protocol, metric code and prompts on a seeded subset of 40 samples per task (520 per method; Table 4). It was run with our initial configuration v1, before the default was fixed. At 2.125 bits v1 averages 46.5 against 42.4 for TurboQuant-MSE^c and 44.8 for KIVI-2 at 3.08 bits; at 3.125 bits it averages 52.8 against 50.9, and fp16 reaches 54.3.

5.3 What each component contributes

Changing the lattice, the coder or the centering costs accuracy, the rounds matter for the rate rather than for perplexity, and every component we removed from our initial design was costing accuracy (Table 5, one change at a time). *Lattice:* replacing E_8 by $\mathbb{Z}$ with everything else fixed costs 2.3% [2.0, 2.5] at 2.125 bits and 0.3% [0.1, 0.4] at 3.125 bits. *Coder:* a Rice coder instead of rANS at equal rate costs 4.0% [3.6, 4.4]. *Centering:* without it, perplexity exceeds 10^4. *Rounds:* with $R = 0$, 1 or 2 instead of 3, perplexity changes by at most 0.3%, and no interval excludes zero; over five seeds the means for $R = 1$, 2 and 3 are 6.936, 6.937 and 6.941. Without rotation, however, the rate misses its target (2.112 bits for 2.125), because the Gaussian code model no longer fits. *Design ablation:* in our initial configuration v1, removing the dither (64 reused rows), the Wiener scaling or the key shaping each improved perplexity at 2.125 bits (1.3%, 0.8%, 0.5%), while replacing E_8 by $\mathbb{Z}$ ($+3.1\%$), rANS by a Rice coder ($+5.7\%$), or removing the centering (perplexity above 8000) each hurt; hence the default. This choice was made on Qwen2.5-7B wikitext2; C4 and Mistral-7B served as held-out checks.

Rounds on real keys and values. Real activations are far easier than the worst case of Theorem 4.1: one round already brings real keys and values to the Gaussian value of every statistic we measure (Figure 7). For keys, the peak-to-average ratio $\mu d = d\|y\|_\infty^2/\|y\|_2^2$ falls from about 18.8 to 7.9, the

Table 5: Ablations on Qwen2.5-7B, wikitext2 perplexity, with measured bits and, for HybridQuant, the paired-bootstrap 95% interval of the relative change. Top: HybridQuant with one component changed. Bottom: the design ablation around our initial configuration v1 (dither with 64 reused rows, Wiener scaling, key shaping), which led to the default.

	2.125 bits			3.125 bits		
Variant	Bits	PPL	Δ% [95% CI]	Bits	PPL	Δ% [95% CI]
HybridQuant (E_8, R=3, rANS, centering)	2.124	6.921		3.124	6.509	
$\mathbb{Z}$ lattice instead of E_8	2.123	7.077	+2.3 [+2.0, +2.5]	3.124	6.526	+0.3 [+0.1, +0.4]
$R = 2$ rounds	2.124	6.935	+0.2 [-0.1, +0.5]	3.124	6.512	+0.0 [-0.1, +0.2]
$R = 1$ round	2.124	6.939	+0.3 [-0.0, +0.5]	3.124	6.511	+0.0 [-0.1, +0.2]
$R = 0$ (no rotation)	2.112	6.937	+0.2 [-0.0, +0.5]	3.112	6.513	+0.1 [-0.1, +0.2]
$\mathbb{Z}$ lattice and $R = 1$ (Rot-ECSQ)	2.123	7.081	+2.3 [+2.0, +2.7]	3.124	6.532	+0.4 [+0.2, +0.5]
Rice coder instead of rANS	2.126	7.198	+4.0 [+3.6, +4.4]	–	–	–
no projection-bias centering	2.124	11226	–	–	–	–
with per-position dither (HybridQuant-G)	2.125	6.998	+1.1 [+0.9, +1.4]	3.126	6.519	+0.1 [+0.0, +0.3]
v1 (dither P=64, MMSE, key shaping)	2.125	7.122		3.126	6.533	
no dither	2.124	7.033	−1.3 [-1.5, -1.0]	3.124	6.530	−0.0 [-0.2, +0.1]
no Wiener (MMSE) scaling	2.125	7.063	−0.8 [-1.1, -0.6]	3.126	6.515	−0.3 [-0.4, -0.2]
no key shaping	2.125	7.089	−0.5 [-0.7, -0.2]	3.126	6.521	−0.2 [-0.3, -0.1]
$\mathbb{Z}$ lattice instead of E_8	2.123	7.340	+3.1 [+2.8, +3.3]	3.124	6.554	+0.3 [+0.2, +0.5]
Rice coder instead of rANS	2.127	7.529	+5.7 [+5.3, +6.1]	3.127	6.549	+0.2 [+0.1, +0.4]
$R = 1$ round	2.125	7.130	+0.1 [-0.1, +0.3]	3.126	6.528	−0.1 [-0.2, +0.1]
cube dither instead of Voronoi	2.125	7.135	+0.2 [-0.1, +0.4]	3.126	6.529	−0.1 [-0.2, +0.1]
best of 4 sign sets per row	2.125	7.139	+0.2 [-0.0, +0.5]	3.126	6.540	+0.1 [-0.1, +0.3]
no projection-bias centering	2.125	9702	–	3.126	8688	–

spread of block RMS within a row from 0.50 to 0.24, and escape symbols disappear; two and three rounds leave these statistics unchanged, as Theorem F.2 predicts for the peak. The third round is therefore insurance rather than a source of accuracy, and it is cheap: on an A100 it adds 0.09 ms, about 6%, to the unfused quantize–dequantize round trip of 4096 tokens $\times$ 8 heads (Table 23).

Designs not adopted. Gaussianising coordinate maps (a Wasserstein-optimal marginal transform via the Cornish–Fisher expansion), learned, PCA and dense rotations, the Leech lattice, nested lattices, Babai decoding, best-of-K sign sets, Rice, Golomb and adaptive coders, Wiener scaling, key shaping and cube dither each need calibration data, add side information or computation without a measurable gain, or break a guarantee (Appendix O).

6 Limitations and conclusion

Limitations. Our kernels are unfused and rANS decoding runs on the CPU, so we make no end-to-end speed or memory-traffic claims; a deployment needs a fused GPU decoder and per-stream offsets (Section 3.3). We evaluate three 7–8B model families with $d = 128$, and not larger models, other head dimensions, KVTC or CacheGen. The third Hadamard round rests on a worst-case analysis whose upper bound is a proof sketch, and it does not measurably change perplexity on the models we test. On Qwen, LongBench-E was run only for the initial configuration v1. The coder of HybridQuant-G ignores the dither value, which wastes up to $I(Q;U)$ bits per block (Appendix B.2).

Conclusion. Calibration-free and entropy-coded KV quantization have been treated as a trade-off. They need not be: once a rotation makes every vector nearly Gaussian, one entropy code built offline for a Gaussian source meets a rate fixed in advance on real models, without seeing their data. How much rotation this takes has a precise answer: one round suffices on the models we test, two provably bring every input close to Gaussian, and a cheap third round lowers the floor that two leave on a spike. With E_8 and static rANS, the result is significantly better than fixed-rate and scalar entropy-coded quantizers at every rate we test; the dithered mode adds provable guarantees at a small cost; and centering exactly removes a failure of rotation-based quantizers on Qwen2 models.

AI USE STATEMENT

In this work, we used generative AI (a large language model used through an agentic coding assistant) to help develop the theoretical framework, formulate mathematical claims, provide critical ingredients for proofs and assist in writing them (in particular the analysis of one, two and three Hadamard rounds, the peak and chaining results and the attention concentration bound), propose and refine hypotheses (for example, that the projection biases of Qwen2 cause the collapse of rotation-based quantizers), design and give feedback on the methodology and experiments (the operating modes, the matched ablations, the rounds sweep and the significance tests), implement the method (the C++/CUDA kernels, the entropy coder, the experiment runners and the tests), generate synthetic data (the outlier, heavy-tailed and unit-sphere inputs used for the MSE, inner-product, rate and Kolmogorov-distance simulations in the appendix), and interpret results, and assisting with translation, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create scientific figures, suggest experimental parameters, create and edit software code and other artifacts, draft parts of the paper, edit it for readability, suggest its structure, brainstorm, search for, identify, summarize and analyse related literature and gaps, and format references. We have reviewed all AI-assisted work. All experiments were run by the authors with the released code; the result tables of the main text are generated programmatically from the stored run records, which prevents transcription errors, and the statistical tests are computed from the same records. The synthetic data are generated by seeded scripts released with the code, so every simulation can be rerun exactly, and they are used only for the appendix analyses, never in place of the real-model evaluations. The code is covered by an automated test suite that includes bit-exact agreement of the CPU and GPU kernels with a reference implementation and a bit-exact round trip of the entropy coder. Every bibliography entry was checked against its primary source (publisher, proceedings, OpenReview or arXiv page), and entries found to be incorrect were corrected by the authors. Claims were checked against the evidence through an independent critical review of the draft, which led us to remove unsupported claims and to add the matched ablations, seeds and significance tests; proofs that are not complete are labelled as proof sketches, and the status of every claim is listed in the appendix. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work concerns memory-efficient inference for existing public language models. It involves no human subjects and no new data collection. It uses public datasets (wikitext2, C4, LongBench) under their licences and releases no model weights. Cheaper inference lowers the cost of deploying language models for both beneficial and harmful uses; our method does not change what a model can do.

REPRODUCIBILITY STATEMENT

Anonymized code is provided as supplementary material: the C++/CUDA kernels, bit-exact CPU/GPU tests, resumable experiment runners, every configuration file, the significance-testing script, and scripts that reproduce every table and figure (Appendix V). All proofs are in the appendix, where each claim is labelled as proved, proof sketch, measured or open (Table 28).

REFERENCES

Nir Ailon and Bernard Chazelle. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009. doi: 10.1137/060673096.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 100213–100240, 2024.

László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6 (1):1–13, 1986. doi: 10.1007/BF02579403.

Mario Bebendorf. A note on the Poincaré inequality for convex domains. *Zeitschrift für Analysis und ihre Anwendungen*, 22(4):751–756, 2003. doi: 10.4171/ZAA/1170.

Witold Bednorz and Rafał Latała. On the boundedness of Bernoulli processes. *Annals of Mathematics*, 180(3):1167–1203, 2014. doi: 10.4007/annals.2014.180.3.8.

Sergey Bobkov and Michel Ledoux. Poincaré's inequalities and Talagrand's concentration phenomenon for the exponential distribution. *Probability Theory and Related Fields*, 107(3):383–400, 1997.

Mark Boss, Vikram Voleti, Simon Donné, and Shimon Vainer. OCTOPUS: Optimized KV cache for transformers via octahedral parametrization under optimal squared error quantization. *arXiv preprint arXiv:2605.21226*, 2026.

Victor Chernozhukov, Iván Fernández-Val, and Alfred Galichon. Rearranging Edgeworth–Cornish–Fisher expansions. *Economic Theory*, 42(2):419–435, 2010. doi: 10.1007/s00199-008-0431-z.

Krzysztof Choromanski, Mark Rowland, and Adrian Weller. The unreasonable effectiveness of structured random orthogonal embeddings. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/hash/bf8229696f7a3bb4700cfddef19fa23f-Abstract.html`.

John H. Conway and Neil J. A. Sloane. Fast quantizing and decoding algorithms for lattice quantizers and codes. *IEEE Transactions on Information Theory*, 28(2):227–232, 1982a. doi: 10.1109/TIT.1982.1056484.

John H. Conway and Neil J. A. Sloane. Voronoi regions of lattices, second moments of polytopes, and quantization. *IEEE Transactions on Information Theory*, 28(2):211–226, 1982b. doi: 10.1109/TIT.1982.1056483.

John H. Conway and Neil J. A. Sloane. A fast encoding method for lattice codes and quantizers. *IEEE Transactions on Information Theory*, 29(6):820–824, 1983. doi: 10.1109/TIT.1983.1056761.

John H. Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften*. Springer, New York, 3rd edition, 1999. doi: 10.1007/978-1-4757-6568-7.

E. A. Cornish and R. A. Fisher. Moments and cumulants in the specification of distributions. *Revue de l'Institut International de Statistique / Review of the International Statistical Institute*, 5(4):307, 1938. doi: 10.2307/1400905.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30318–30332, 2022.

Yuval Domb, Hadar Sackstein, and Tomer Solberg. HyperQuant: A rate-distortion-optimal quantization pipeline for large language and diffusion models. *arXiv preprint arXiv:2606.23406*, 2026.

Jarek Duda. Asymmetric numeral systems: Entropy coding combining speed of Huffman coding with compression rate of arithmetic coding. *arXiv preprint arXiv:1311.2540*, 2013. doi: 10.48550/arXiv.1311.2540.

Richard M. Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967. doi: 10.1016/0022-1236(67)90017-1.

William Feller. *An Introduction to Probability Theory and Its Applications*, volume II. Wiley, New York, 2nd edition, 1971.

Robert M. Gray and David L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383, 1998. doi: 10.1109/18.720541.

Mikhael Gromov and Vitali D. Milman. A topological application of the isoperimetric inequality. *American Journal of Mathematics*, 105(4):843–854, 1983. doi: 10.2307/2374298.

Uffe Haagerup. The best constants in the Khintchine inequality. *Studia Mathematica*, 70(3):231–283, 1981. doi: 10.4064/sm-70-3-231-283.

Insu Han, Praneeth Kacham, Amin Karbasi, Vahab Mirrokni, and Amir Zandieh. PolarQuant: Vector quantization with polar transformation. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 5113–5121. PMLR, 2026. URL https://proceedings.mlr.press/v300/han26c.html.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 1270–1303, 2024. doi: 10.52202/079017-0040.

Rafał Latała. Estimates of moments and tails of Gaussian chaoses. *The Annals of Probability*, 34 (6):2315–2331, 2006. doi: 10.1214/009117906000000421.

Michel Ledoux. *The Concentration of Measure Phenomenon*, volume 89 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2001.

Junyan Li, Yang Zhang, Muhammad Yusuf Hassan, Talha Chafekar, Tianle Cai, Zhile Ren, Pengsheng Guo, Foroozan Karimzadeh, Colorado Reed, Chong Wang, and Chuang Gan. CommVQ: Commutative vector quantization for KV cache compression. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 36831–36845. PMLR, 2025.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? An empirical study on quantized reasoning models. In *Conference on Language Modeling (COLM)*, 2025a.

Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, Michael Maire, Henry Hoffmann, Ari Holtzman, and Junchen Jiang. CacheGen: KV cache compression and streaming for fast large language model serving. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pp. 38–56. ACM, 2024a. doi: 10.1145/3651890.3672274.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025b.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR, 2024b.

João Marques. NexusQuant: Training-free E8 lattice vector quantization for KV cache compression. Technical report, 2026. URL https://github.com/jagmarques/nexusquant/blob/main/paper/nexusquant-method.pdf.

Or Ordentlich and Yury Polyanskiy. Optimal quantization for matrix multiplication. *IEEE Transactions on Information Theory*, 72(3):1943–1972, 2026. doi: 10.1109/TIT.2025.3649596. Theorem, proposition and equation numbers cited here refer to arXiv:2410.13780v3.

Lawrence E. Payne and Hans F. Weinberger. An optimal Poincaré inequality for convex domains. *Archive for Rational Mechanics and Analysis*, 5(1):286–292, 1960. doi: 10.1007/BF00252910.

Valentin V. Petrov. *Sums of Independent Random Variables*, volume 82 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, Berlin, Heidelberg, 1975. doi: 10.1007/978-3-642-65809-9.

John K. Salmon, Mark A. Moraes, Ron O. Dror, and David E. Shaw. Parallel random numbers: As easy as 1, 2, 3. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–12, 2011. doi: 10.1145/2063384.2063405.

Semyon Savkin, Eitan Porat, Or Ordentlich, and Yury Polyanskiy. NestQuant: Nested lattice quantization for matrix products and LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53042–53062. PMLR, 2025.

Utkarsh Saxena and Kaushik Roy. KVLinC: KV cache quantization with Hadamard rotation and linear correction. *arXiv preprint arXiv:2510.05373*, 2025.

Irina G. Shevtsova. An improvement of convergence rate estimates in the Lyapunov theorem. *Doklady Mathematics*, 82(3):862–864, 2010. doi: 10.1134/S1064562410060062.

Konrad Staniszewski and Adrian Łańcucki. KV cache transform coding for compact storage in LLM inference. In *The Fourteenth International Conference on Learning Representations*, 2026. arXiv:2511.01815.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *Conference on Language Modeling*, 2024.

Michel Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'IHÉS*, 81:73–205, 1995. doi: 10.1007/BF02699376.

Michel Talagrand. *Upper and Lower Bounds for Stochastic Processes: Decomposition Theorems*, volume 60 of *Ergebnisse der Mathematik und ihrer Grenzgebiete, 3. Folge*. Springer, Cham, 2nd edition, 2021. doi: 10.1007/978-3-030-82595-9.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. doi: 10.1007/s10208-011-9099-z.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with Hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48630–48656. PMLR, 2024.

Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. *Advances in Neural Information Processing Systems*, 29, 2016.

Ram Zamir. *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation and Multiuser Information Theory*. Cambridge University Press, Cambridge, 2014. ISBN 9780521766982.

Ram Zamir and Meir Feder. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory*, 38(2):428–436, 1992. doi: 10.1109/18.119699.

Ram Zamir and Meir Feder. On lattice quantization noise. *IEEE Transactions on Information Theory*, 42(4):1152–1159, 1996. doi: 10.1109/18.508838.

Amir Zandieh, Majid Daliri, and Insu Han. QJL: 1-bit quantized JL transform for KV cache quantization with zero overhead. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25805–25813, 2025.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online vector quantization with near-optimal distortion rate. In *International Conference on Learning Representations*, volume 2026, pp. 56418–56439, 2026.

Jintao Zhang, Jia Wei, Pengle Zhang, Jun Zhu, and Jianfei Chen. SageAttention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations*, volume 2025, pp. 71566–71585, 2025.

Tianyi Zhang, Jonah Yi, Zhaozhuo Xu, and Anshumali Shrivastava. KV cache is 1 bit per channel: Efficient large language model inference with coupled quantization. In *Advances in Neural Information Processing Systems*, volume 37, pp. 3304–3331, 2024. doi: 10.52202/079017-0109.

A Overview of the appendix

This appendix contains the complete mathematical analysis of HybridQuant (identifier `e8kv` in the released code) and all supporting evidence. HybridQuant has two operating modes (Appendix B.1). The default, accuracy mode, called HybridQuant, uses $R = 3$ same-axis randomized Hadamard rounds, the E_8 lattice, static rANS coding and projection-bias centering, without dither. The guaranteed mode HybridQuant-G adds subtractive Voronoi dither with an independent dither row for every position. We also refer to the initial configuration v1, the starting point of the design ablation. The results on the rotation hold in every mode; the distortion and attention guarantees (Theorems H.2 and I.4) need the dither and hold for HybridQuant-G.

Every statement is labelled by its status: Proved (a complete proof is given or a standard theorem is applied with all hypotheses verified), Proof sketch (the argument is complete except for a routine local-limit estimate whose constants we certify numerically), Measured (an exact or Monte-Carlo computation with the released code), or Open. Appendix W lists every claim with its status. We separate carefully what the rotation provably does, what it does not do, and what is only observed.

Main conclusions.

1. *With dither, distortion does not depend on anisotropy.* With per-row RMS scaling and subtractive Voronoi dither, as in HybridQuant-G, the reconstruction error of every row is uniform on the scaled E_8 Voronoi cell and independent of the input, for every number of Hadamard rounds $R \geq 0$ (Theorem H.2). Anisotropy of the input enters only the *rate* (escape symbols and the gap between design and measured rate). Without dither, as in the default HybridQuant, the error depends on the input and the rounds matter (Table 14).
2. *One round is not enough for a calibration-free design.* After one round the distribution seen by the lattice depends on the input: a spike becomes a two-point law at Kolmogorov distance 0.3413 from $\mathcal{N}(0,1)$ for every d (Theorem E.7). For $R \geq 2$ the Berry–Esseen distance is at most $0.56\sqrt{2\log(2d/\delta)/d}$ for *every* input with probability $1-\delta$, and at most $0.894/\sqrt{d}$ in expectation (Theorem E.7). One Sylvester round also has exact, dimension-free block collisions (Lemma F.7); a second round removes them up to a logarithmic factor (Lemma F.8).
3. *Three rounds improve on two.* For a sparse outlier, the two-round output is supported on a lattice of spacing $2\sigma/\sqrt{d}$ and its Kolmogorov distance to $\mathcal{N}(0,1)$ is at least $\frac{1}{2}\binom{d}{d/2}2^{-d} \sim (2\pi d)^{-1/2}$ (Proposition G.1). After three rounds it is $\Theta(d^{-1})$ up to a $\log d$ factor (Theorem G.2; the lower bound is proved, the upper bound is a proof sketch). The exact values are 0.0352 versus 0.0097 at $d = 128$; the ratio grows as $\sqrt{d}$ (Table 16).
4. *What three rounds do not do.* They do not reduce the peak $\|y\|_\infty$: for sparse outliers one round gives $\|y\|_\infty \leq \sqrt{k}\,\sigma$, while any $R \geq 2$ gives $\mathbb{E}\|y\|_\infty \gtrsim \sigma\sqrt{\log d}$ (Theorem F.2). They do not change the dithered distortion (Theorem H.2).
5. *Attention is stable at long context.* The attention output is Lipschitz in the key and value quantization errors with a constant that does not depend on the context length T (Lemma I.2), in every mode. In HybridQuant-G, whose dithers are independent across positions for contexts of up to 40960 tokens, its variance is bounded by a T-independent constant (Theorem I.4).
6. *Rates are counted and predictable.* The reported rate counts every side bit (Appendix B). On real models the effective rate is within 0.003 bits of the target in every run with $R \geq 1$; with $R = 0$, the rate averaged over keys and values deviates by up to 0.012 bits (Appendix B.2).

Organisation. Appendix B specifies the quantizer, its bit accounting, the operating modes (Appendix B.1) and the entropy coder (Appendix B.2). Appendix C covers lattice quantization with dither, the SNR knob and MMSE scaling. Appendix D derives the data-free RoPE-pair key shaping used in v1. Appendix E proves the distributional properties of R Hadamard rounds, Appendix F the peak and coherence results, and Appendix G the two-versus-three-round separation. Appendix H shows where anisotropy does and does not enter the pipeline, Appendix I gives the attention-level guarantees, Appendix K relates the design to the rate–distortion theory of matrix multiplication, and Appendix L covers projection-bias centering and the closest related work. Appendix P reports the numerical evidence, Appendix Q the statistical methodology and the baseline implementations, Appendix U summarises the case for three rounds, Appendix V documents reproducibility, and Appendix W is the status ledger. The experiments of this version are specified in `configs/final.yaml` and run by `scripts/final_runs.sh`; the statistical tests are computed by `e8kv/significance.py`.

Notation. d is the head dimension, a power of two with $8 \mid d$; T is the context length. $H \in \mathbb{R}^{d\times d}$ is the normalised Sylvester–Hadamard matrix, $H_{ij} = d^{-1/2}(-1)^{\langle i,j\rangle}$ with $\langle i,j\rangle$ the bitwise inner product mod 2, and $W = \sqrt{d}\,H \in \{\pm 1\}^{d\times d}$. $D_r = \mathrm{diag}(\epsilon_r)$ with $\epsilon_1, \dots, \epsilon_R$ independent Rademacher vectors. The R-round transform is $Q_R = HD_R \cdots HD_1$ and $y^{(r)} = HD_r y^{(r-1)}$, $y^{(0)} = x$. For $z \in \mathbb{R}^d$, $\sigma(z) = \|z\|_2/\sqrt{d}$ is its RMS and $\mu(z) = \|z\|_\infty^2/\|z\|_2^2 \in [1/d, 1]$ its anisotropy. Φ is the standard normal CDF and $d_{\mathrm{K}}(F,G) = \sup_t |F(t) - G(t)|$ the Kolmogorov distance. Λ denotes a lattice, $\mathcal{V}$ its Voronoi cell and Q_Λ the nearest-point map. $G(\Lambda)$ is the normalised second moment (Definition C.1). $\log$ is natural unless stated otherwise.

B THE QUANTIZER AND ITS BIT ACCOUNTING

Algorithm 1 is the exact procedure implemented in the released code (`e8kv/quantizers.py`, kernels in `csrc/e8_core.h`). Keys are quantized immediately after RoPE and values at the output of W_V, each token once, before any query attends to it. On models with key and value projection biases, x is the centered row of Appendix L.1. All randomness (sign vectors, dither tables, Monte-Carlo code models) comes from CPU generators seeded by SHA-256 of $(\text{seed}, \text{layer}, \text{role}, \dots)$, so it is identical on every machine. The dither, the Wiener (MMSE) factor and the key shaping are optional; Appendix B.1 lists which of them each operating mode uses.

Algorithm 1 E8 KV quantization of one row $x \in \mathbb{R}^d$ (one token, one KV head)

Require: row x, role $\in \{K, V\}$, layer ℓ, position t; sign vectors $\epsilon_1, \dots, \epsilon_R \in \{\pm 1\}^d$; relative step $s > 0$. Optional: data-free shaping vector $c \in \mathbb{R}^d_{>0}$ (keys only, Appendix D; otherwise $c = \mathbf{1}$); dither table $u_0, \dots, u_{P-1} \in \mathbb{R}^d$ with period P (otherwise $u \equiv 0$); Wiener factor $\alpha \in (0, 1]$ (otherwise $\alpha = 1$)

$z \leftarrow c \odot x$ **if** role $= K$ **else** x {RoPE-pair shaping}
$y \leftarrow HD_R \cdots HD_1\, z$ {R same-axis Hadamard rounds, $O(Rd \log d)$}
$\hat\sigma \leftarrow \mathrm{fp16}\big(\|y\|_2/\sqrt{d}\big)$, $\Delta \leftarrow \hat\sigma s$ {16 side bits per row}
for each 8-block b **do**
  $\lambda_b \leftarrow Q_{E_8}\big(y_b/\Delta + u_{t \bmod P, b}\big)$ {exact Conway–Sloane decoder}
end for
entropy-code the λ_b with a static context-switched rANS model (Appendix B.2)
$\hat y_b \leftarrow \alpha\Delta\,(\lambda_b - u_{t \bmod P, b})$ {subtractive dither, Wiener scaling}
$\hat x \leftarrow c^{-1} \odot (HD_1)^\top \cdots (HD_R)^\top \hat y$ (keys) or $(HD_1)^\top \cdots (HD_R)^\top \hat y$ (values)

Step size. The relative step s is set in one of two data-free ways. With the *SNR knob*, $s = 1/\sqrt{G_{E_8} L}$ with $L = 10^{\mathrm{SNR}/10} - \mathbf{1}[\alpha < 1]$ (Propositions C.5 and C.6). With a *rate target*, s solves "rate of the static code on an i.i.d. $\mathcal{N}(0,1)$ source = target payload" by 30 bisection steps on $\log s$ (Monte-Carlo with 2^{16} blocks); the code model is then built with 2^{20} Gaussian blocks (Appendix B.2). All experiments on real models use rate targets; the synthetic experiments of Appendix P use the SNR knob. No calibration data are used anywhere.

Bit accounting. The reported rate is the *effective* number of bits per element and includes every side bit:

$$\text{bits/elem} = \underbrace{\tfrac{1}{n}\textstyle\sum_{\text{symbols}} \ell(\text{symbol})}_{\text{rANS payload}} + \underbrace{16/d}_{\text{fp16 RMS}} + \underbrace{32/4096}_{\text{stream flush}} + \underbrace{\mathbf{1}[K_{\text{sel}} > 1]\log_2 K_{\text{sel}}/d}_{\text{best-of-}K}, \qquad (1)$$

where ℓ is the code length of the static model (escape symbols cost the escape code plus 16 raw bits). For $d = 128$ the side information is $0.125 + 0.0078 = 0.133$ bits per element. Appendix B.2 specifies the code model and describes how this accounting is checked against real rANS bitstreams.

Where the implementation deviates from the idealised analysis. The proofs below idealise four details; we state them here once.

1. *Dither period.* The analysis assumes an independent dither for every (position, block). The implementation draws, for every (layer, role), a table of P dither rows from the seed and uses row $t \bmod P$ at position t. Each row consists of $d/8$ independent block dithers, keys and values have separate tables, and all KV heads of a layer share the row of their position. In HybridQuant-G, $P = 40960$ exceeds the length of every sequence evaluated, so no row is reused within a sequence and the independence hypothesis holds exactly. The initial configuration v1 used $P = 64$: blocks within any window of 64 consecutive positions have independent dithers, but positions t and $t + 64$ share one. Results that need independence *across* positions (Theorem I.4) then apply exactly only for $T \le 64$; per-row results (Theorems C.4 and H.2 and Proposition C.5) do not depend on P. The default HybridQuant has no dither.
2. *fp16 RMS.* The scale uses $\hat{\sigma} = \text{fp16}(\sigma)$, $|\hat{\sigma}/\sigma - 1| \le 2^{-11}$. All per-row identities hold with $\hat{\sigma}$ in place of σ.
3. *Wiener scaling.* With $\alpha < 1$, which only v1 uses, the error is $(\alpha - 1)y + \alpha e$. HybridQuant and HybridQuant-G use $\alpha = 1$; we state results for $\alpha = 1$ and note the modification where it matters.
4. *Kernel arithmetic.* Kernels use float32 add/sub/mul only, round-half-to-even and disabled FMA contraction, so the CPU kernel, the GPU kernel and the PyTorch reference agree bit for bit. Even if the rotation were computed entirely in fp16 arithmetic, its round-off would be more than four orders of magnitude below the quantization noise for $R \le 3$ (Table 19).

B.1 Operating modes

Table 6 lists the three configurations that appear in this paper. All three use the E_8 lattice, $R = 3$ same-axis rounds, an fp16 RMS per row, static rANS coding with a rate target (Appendix B.2) and projection-bias centering (Appendix L.1). They differ in the dither, the Wiener factor and the key shaping.

Table 6: Operating modes. HybridQuant is the default (accuracy) mode, HybridQuant-G the guaranteed mode, and v1 the initial configuration of the design ablation. Configuration names are those of `configs/final.yaml` (`a3-`$\star$, `gp-`$\star$) and `configs/main.yaml` (`ours-`$\star$). Shaping: RoPE-pair key shaping (Appendix D). The last two columns state whether, and for which context lengths T, the hypotheses of Theorem H.2 and Theorem I.4 hold.

Mode	Configs	Dither, period P	α	Shaping	R	Thm. H.2	Thm. I.4
HybridQuant	`a3-`$\star$	none	1	no	3	no	no
HybridQuant-G	`gp-`$\star$	Voronoi, $P = 40960$	1	no	3	yes	$T \le 40960$
v1	`ours-`$\star$	Voronoi, $P = 64$	$1/(1 + G_{E_8} s^2)$	yes	3	no ($\alpha < 1$)	$T \le 64$

For v1, $\alpha = 1/(1 + G_{E_8} s^2)$ is about 0.92 at 2.125 bits and 0.98 at 3.125 bits, with s from the rate solver of Appendix B.2. The design ablation of the main text (Table 5) started from v1 and changed one component at a time. At 2.125 bits, removing the dither, the Wiener scaling or the key shaping each lowered perplexity, while replacing E_8 by $\mathbb{Z}$ or rANS by a Rice coder, or removing the centering, each raised it. The default HybridQuant therefore keeps E_8, rANS and centering and drops the other three. HybridQuant-G restores only the dither, because Theorems H.2 and I.4 need it, and uses one dither row per position so that no row is reused within any sequence evaluated: needle

prompts have at most 32768 tokens, LongBench prompts are truncated to 31500 tokens before the chat template is applied, and at most 512 tokens are generated.

The results on the rotation (Theorems E.7, F.1, F.2 and G.2 and Propositions F.3 and G.1) and Lemma H.1 and Proposition L.1 hold in all three modes. Theorem H.2 needs dither and $\alpha = 1$ and therefore holds for HybridQuant-G only. Theorem I.4 needs, in addition, independent dithers for all (position, block) pairs; HybridQuant-G provides them for contexts of up to 40960 tokens, and v1 provided them only for $T \leq 64$. Lemma I.2 and Proposition I.3 use only that every block error lies in its scaled Voronoi cell, which holds for nearest-point quantization with or without dither, so they hold for HybridQuant and HybridQuant-G.

B.2 Entropy coder specification

This subsection states what the released code does (`e8kv/entropy.py`, `e8kv/quantizers.py`, `csrc/rans.h`, `csrc/e8_core.h`).

Symbols. The kernels store a lattice point λ as its doubled coordinates $c = 2\lambda \in \mathbb{Z}^8$ in int16. For E_8 all coordinates of c have the same parity $\nu \in \{0, 1\}$, which identifies the coset (D_8 or $D_8 + \frac{1}{2}\mathbf{1}$), and $w = (c - \nu\mathbf{1})/2 \in D_8$. A block is coded as nine symbols: the coset bit ν; the integers $w_1, \ldots, w_7$; and $(w_8 - \pi)/2$ with $\pi = (w_1 + \cdots + w_7) \bmod 2$. Because the coordinates of $w \in D_8$ have an even sum, $w_8 \equiv \pi \pmod 2$, so the last symbol is an integer and the parity of w_8 is not coded. For $\mathbb{Z}$ every coordinate $c_i/2$ is one symbol. In every context a value $v \in [-A, A]$ is coded directly; any other value is coded as an escape symbol followed by the value in 16 raw bits, so each context has an alphabet of $2A + 2$ symbols. Here $A = \min(2047, \max(4, 1 + m))$, where m is the largest $|v|$ in the Monte-Carlo sample from which the table is built, so escapes occur only for values that this Gaussian sample never produced.

Contexts. E_8 uses seven contexts: one for ν, one for each value of ν for the symbols $w_1, \ldots, w_7$, and one for each pair (ν, π) for the last symbol. Each context is a function of symbols decoded earlier in the same block, so a decoder can track it. $\mathbb{Z}$ uses a single context. The tables are static: nothing adapts across blocks, rows, tokens or layers.

Probabilities and coder. Symbol counts are smoothed by adding $\frac{1}{2}$ and quantized deterministically to integer frequencies $f \geq 1$ that sum to 2^{15} in every context (`prob_bits` $= 15$), so every symbol, including the escape, can be coded in every context. A symbol of frequency f costs $\ell = \log_2(2^{15}/f)$ bits. The coder is a static rANS (Duda, 2013) with a 32-bit state, byte-wise renormalisation and lower bound 2^{23}. Each stream of 4096 elements also stores the coder's final 4-byte state, which is the 32-bit flush of equation 1.

Building the tables. For a lattice, a dither law and a step s, the table is built from $n_{\text{blocks}} = 2^{20}$ blocks of i.i.d. $\mathcal{N}(0, 1)$ coordinates, i.e. rows of RMS 1, quantized as in Algorithm 1 with $\Delta = s$. In a dithered mode every Monte-Carlo block gets a fresh dither from the same law as in deployment (Lemma C.3); in an undithered mode it gets none. The Gaussian source and the dithers come from generators seeded by the global seed. The rotation is not simulated: the table models the Gaussian law that the rotated coordinates approximate (Theorem E.7). No model data are used.

Rate targeting. A configuration specifies an effective rate b in bits per element. The side information is 16 bits of fp16 RMS per row (plus $\log_2 K$ bits with best-of-K sign sets) and a 32-bit flush per 4096 elements, $16/d + 32/4096 = 0.1328$ bits per element at $d = 128$. The step s is the solution of "Monte-Carlo code rate $= b - 0.1328$", found by 30 bisection steps on $\log s \in [\log 10^{-3}, \log 20]$; each step builds a table from 2^{16} Gaussian blocks. The final table is then built at this s, rounded to float32, from 2^{20} blocks. The rate target is therefore met by a data-free numerical search, not in closed form; the SNR knob (Propositions C.5 and C.6) is the closed-form alternative.

How rates are measured. The rate reported for every experiment is given by equation 1: the static-model code length of the actual codes, including 16 raw bits per escape, plus the side bits, with the flush prorated at $32/4096$ bits per element. Evaluation computes these code lengths and does not produce a bitstream. `e8kv/verify_rans.py` checks the accounting against real bitstreams.

It runs one window of a model through the quantizer (2048 tokens of WikiText-2 by default) and captures the integer codes of all layers. For keys and for values separately, it splits the symbol sequence into streams of 4096 elements (4608 symbols for E_8), encodes every stream with the C++ coder of `csrc/rans.h`, decodes it, reinserts the escaped values, and checks that the decoded codes equal the original codes bit for bit; it aborts otherwise. It then reports the actual size (8 times the stream bytes, plus 16 bits per escape) against the accounted size (the static-model code length plus 32 bits per stream) and their relative gap. In this check, escaped values are counted at 16 bits but carried outside the rANS streams. On Qwen2.5-7B and Mistral-7B, at 2.125 and 3.125 bits, every stream decodes bit-exactly and the real bitstreams are 0.03–0.05% smaller than the accounted size.

Across all runs of our pipeline on real models, the effective rate is within 0.003 bits of the target for every configuration with $R \geq 1$ (2.123–2.128 bits for a 2.125-bit target, 3.124–3.128 for 3.125). With $R = 0$ the rate, averaged over keys and values, deviates by up to 0.012 bits (2.113 at a 2.125-bit target without dither, 3.116 at 3.125), consistent with the synthetic results of Table 13: the Gaussian code model fits rotated rows but not unrotated ones.

The code model ignores the dither value. In HybridQuant-G the tables are built from dithered Monte-Carlo codes, but neither the tables nor the contexts depend on the value of the dither, which the decoder knows from the shared seed. Up to the approximation made by the context model, the coder therefore spends the marginal entropy $H(Q)$ of the lattice index Q of a block. A code conditioned on the dither U could approach $H(Q \mid U)$, which for subtractive dither equals $I(X; X + E)$ with E uniform on the cell (Zamir & Feder, 1992). The excess is $H(Q) - H(Q \mid U) = I(Q; U) \geq 0$ bits per block. It vanishes in the high-resolution limit; we have not measured it at the rates used here. Because the rate is targeted, any excess is paid for with a larger step, that is, with more distortion. This is a known inefficiency of HybridQuant-G, and it is one plausible reason why the dithered mode costs accuracy at matched rate (Table 5). We state it as a hypothesis: we have not tested it, and we have not implemented a dither-conditioned code. The default HybridQuant has no dither and is not affected.

C Lattice quantization with subtractive dither

C.1 Lattices, the normalised second moment and E_8

Definition C.1 (Normalised second moment)**.** For a lattice $\Lambda \subset \mathbb{R}^n$ with Voronoi cell $\mathcal{V}$,

$$G(\Lambda) = \frac{1}{n} \frac{\int_{\mathcal{V}} \|v\|^2 \, dv}{\mathrm{vol}(\mathcal{V})^{1+2/n}}.$$

G is invariant under scaling and rotation of Λ. It equals the per-coordinate mean squared error of a uniform point in a cell of unit volume.

Known values (Conway & Sloane, 1982b), (Conway & Sloane, 1999, Ch. 2, Ch. 21): $G(\mathbb{Z}^n) = 1/12 \approx 0.083333$, $G_{E_8} = 929/12960 \approx 0.071682$, and every lattice in every dimension has $G > 1/(2\pi e) \approx 0.058550$. E_8 therefore has a granular gain of $10 \log_{10}\big(\frac{1/12}{G_{E_8}}\big) = 0.654$ dB over scalar quantization, out of the 1.533 dB that is possible at all.

We use $E_8 = D_8 \cup (D_8 + \frac{1}{2}\mathbf{1})$ with $D_8 = \{z \in \mathbb{Z}^8 : \sum_i z_i \equiv 0 \pmod 2\}$. This lattice has $\mathrm{vol}(\mathcal{V}) = 1$, minimum norm $\sqrt{2}$ and covering radius 1 (Conway & Sloane, 1999). The exact nearest point is found by the Conway–Sloane decoder (Conway & Sloane, 1982a). To decode D_8, round every coordinate; if the sum is odd, re-round the coordinate with the largest rounding error in the other direction. $Q_{E_8}(v)$ is then the closer of $Q_{D_8}(v)$ and $Q_{D_8}(v - \frac{1}{2}\mathbf{1}) + \frac{1}{2}\mathbf{1}$. The kernels store $2Q_{E_8}(v) \in \mathbb{Z}^8$ as int16.

Lemma C.2 (Isotropy of the E_8 cell)**.** *If* $e \sim \mathrm{Unif}(\mathcal{V}_{E_8})$ *then* $\mathbb{E}e = 0$ *and* $\mathbb{E}ee^\top = G_{E_8} I_8$.

Proof. $\mathcal{V}$ is centrally symmetric, so $\mathbb{E}e = 0$. The automorphism group of E_8 (the Weyl group $W(E_8)$) maps $\mathcal{V}$ onto itself and acts irreducibly on $\mathbb{R}^8$. Hence $C = \mathbb{E}ee^\top$ commutes with this action, and by Schur's lemma $C = cI_8$. Taking traces, $8c = \mathbb{E}\|e\|^2 = 8G_{E_8}$ because $\mathrm{vol}(\mathcal{V}) = 1$. □

Lemma C.3 (The dither sampler is exact). *Let $z \sim \mathrm{Unif}([0,2)^8)$ and $u = z - Q_{E_8}(z)$. Then $u \sim \mathrm{Unif}(\mathcal{V}_{E_8})$.*

Proof. $2\mathbb{Z}^8 \subset D_8 \subset E_8$, and $[0,2)^8$ is a fundamental domain of $2\mathbb{Z}^8$. The uniform law on a fundamental domain of a sublattice pushes forward to the Haar (uniform) law on the torus $\mathbb{R}^8/2\mathbb{Z}^8$. Under the quotient map $\mathbb{R}^8/2\mathbb{Z}^8 \to \mathbb{R}^8/E_8$ this becomes the Haar law on $\mathbb{R}^8/E_8$. The map $z \mapsto z - Q_{E_8}(z)$ picks the representative of $z \bmod E_8$ in $\mathcal{V}$; this representative is unique outside a null set of ties. Hence u is uniform on $\mathcal{V}$. □

By contrast, the "cube" dither $\mathrm{Unif}([-\frac{1}{2},\frac{1}{2})^8)$ has the right volume but is not a fundamental domain of E_8: its translate by $\frac{1}{2}\mathbf{1} \in E_8$ overlaps it on $[0,\frac{1}{2})^8$. Theorem C.4 below therefore does not apply to it, and we keep it only as an ablation. (The proof of Theorem C.4 only uses that $u \bmod \Lambda$ is Haar-distributed, so a dither uniform on *any* fundamental domain of E_8 gives the same error law; Lemma C.3 produces one on $\mathcal{V}$.)

C.2 Subtractive dither: exact error law

Theorem C.4 (Zamir–Feder (Zamir & Feder, 1992; 1996)). *Let X be any random vector in $\mathbb{R}^n$ and $u \sim \mathrm{Unif}(\mathcal{V})$ independent of X. Let $e = Q_\Lambda(X+u) - u - X$. Then $e \sim \mathrm{Unif}(\mathcal{V})$ and e is independent of X. If $X = (X_1,\dots,X_m)$ is quantized blockwise with independent dithers $u_1,\dots,u_m$, the block errors $e_1,\dots,e_m$ are i.i.d. $\mathrm{Unif}(\mathcal{V})$ and jointly independent of X.*

Proof. Condition on $X = x$. Translation by x preserves the Haar measure of the torus $\mathbb{R}^n/\Lambda$, so $(x+u) \bmod \Lambda$ is Haar-distributed. Its representative in $\mathcal{V}$, $(x+u) - Q_\Lambda(x+u) = -e$, is therefore uniform on $\mathcal{V}$. $\mathcal{V}$ is symmetric, so $e \sim \mathrm{Unif}(\mathcal{V})$. This conditional law does not depend on x, which gives independence. For blocks, apply the same argument to the product torus with the product dither. □

The theorem needs no assumption on the input distribution: spikes, heavy tails and correlated channels are all covered. It is the reason the SNR knob below is exact rather than asymptotic.

C.3 The SNR knob and Wiener scaling

Fix a row $y \in \mathbb{R}^d$ with RMS σ, stored RMS $\hat\sigma$, relative step s and cell scale $\Delta = \hat\sigma s$. Without Wiener scaling the reconstruction is $\hat y = y + \Delta e$, where $e = (e_1,\dots,e_{d/8})$ are the block errors of Theorem C.4 in unit-cell coordinates.

Proposition C.5 (SNR knob, exact). *For every row $y \neq 0$ and every input distribution,*

$$\frac{\mathbb{E}_u\|\hat y - y\|^2}{\|y\|^2} = G_{E_8}\, s^2 \Big(\frac{\hat\sigma}{\sigma}\Big)^2, \qquad \mathbb{E}_u(\hat y - y)(\hat y - y)^\top = G_{E_8}\Delta^2 I_d.$$

With $s = (G_{E_8}\, 10^{\mathrm{SNR}/10})^{-1/2}$ the per-row SNR equals the target up to $|20\log_{10}(\hat\sigma/\sigma)| \le 0.0043$ dB.

Proof. Apply Theorem C.4 and Lemma C.2 blockwise. There are $d/8$ blocks, each with $\mathbb{E}\|e_b\|^2 = 8G_{E_8}$, so $\mathbb{E}\|\hat y - y\|^2 = \Delta^2 d G_{E_8}$ and $\|y\|^2 = d\sigma^2$. □

Proposition C.6 (Wiener scaling). *Let $\hat y_\alpha = \alpha(y + \Delta e)$. Then $\mathbb{E}\|\hat y_\alpha - y\|^2 = (1-\alpha)^2\|y\|^2 + \alpha^2 d G_{E_8}\Delta^2$. This is minimised at $\alpha^\star = 1/(1 + G_{E_8}s^2(\hat\sigma/\sigma)^2)$, where the SNR is $1 + 1/(G_{E_8}s^2)$ up to the fp16 factor. When Wiener scaling is enabled (only in v1, Appendix B.1), the implementation uses $\alpha = 1/(1+G_{E_8}s^2)$, and for a target SNR sets $s = (G_{E_8}(10^{\mathrm{SNR}/10}-1))^{-1/2}$. At a fixed s this gains $10\log_{10}(1+G_{E_8}s^2)$ dB, e.g. 0.27 dB at a design SNR of 12 dB.*

Proof. The cross term vanishes because $\mathbb{E}e = 0$ and e is independent of y (Theorem C.4). Minimise the quadratic in α. □

Remark C.7. With $\alpha < 1$ the error $(\alpha-1)y + \alpha\Delta e$ is correlated with the signal. Logits are shrunk by α, a deterministic effect that attention partly absorbs. We therefore state the attention results for $\alpha = 1$.

C.4 WHY THE STEP IS CHOSEN FROM THE VARIANCE ALONE

A Cornish–Fisher grid would correct quantile levels using the estimated skewness γ_1 and excess kurtosis γ_2 of the rotated coordinates. The next proposition shows that after one or more rounds these corrections are zero or $O(1/d)$, which is below their own estimation noise.

Proposition C.8 (Moments after rotation)**.** *Fix $r \geq 1$ and condition on $z = y^{(r-1)}$. Every coordinate $y_i^{(r)}$ has a symmetric law, so all its odd moments vanish and $\gamma_1 = 0$ exactly. Its excess kurtosis is exactly*

$$\gamma_2\big(y_i^{(r)} \mid z\big) = -2\,\frac{\|z\|_4^4}{\|z\|_2^4} \in \Big[-2, -\frac{2}{d}\Big].$$

For $r \geq 2$ and any input, $\mathbb{E}\,\gamma_2 \geq -6/d$.

Proof. $y_i = \sum_j a_j \epsilon_{r,j}$ with $a_j = H_{ij} z_j$, and $(\epsilon_{r,j}) \overset{d}{=} (-\epsilon_{r,j})$ gives symmetry. Cumulants of independent sums add, and the fourth cumulant of a Rademacher variable is $\mathbb{E}\epsilon^4 - 3 = -2$. Hence $\kappa_4 = -2\sum_j a_j^4 = -2\|z\|_4^4/d^2$ and $\kappa_2^2 = \|z\|_2^4/d^2$. For $r \geq 2$, $z = HD_{r-1}w$, and Khintchine's inequality with Haagerup's constant (Haagerup, 1981), $\mathbb{E}z_j^4 \leq 3(\|w\|^2/d)^2$, gives $\mathbb{E}\|z\|_4^4 \leq 3\|w\|^4/d = 3\|z\|_2^4/d$. □

For comparison, the sample excess kurtosis of d near-Gaussian coordinates has standard error $\approx \sqrt{24/d}$, which is 0.43 at $d = 128$, whereas the quantity it estimates is at most $6/d = 0.047$ in expectation. A per-row Cornish–Fisher correction would therefore mostly fit noise. The variance-only SNR knob uses the one statistic that is estimated exactly per row, namely $\hat{\sigma}$.

C.5 EXACT DECODING VERSUS BABAI'S ALGORITHM

Proposition C.9 (Babai cells are boxes)**.** *Let B be any basis of a lattice $\Lambda \subset \mathbb{R}^n$ with Gram–Schmidt vectors $b_1^*, \ldots, b_n^*$. Babai's nearest-plane algorithm (Babai, 1986) partitions $\mathbb{R}^n$ into translates of the box $\{\sum_i c_i b_i^* : c_i \in [-\frac{1}{2}, \frac{1}{2})\}$. The normalised second moment of this partition is*

$$G_{\text{Babai}} = \frac{1}{12n}\,\frac{\sum_i \|b_i^*\|^2}{\big(\prod_i \|b_i^*\|\big)^{2/n}} \;\geq\; \frac{1}{12},$$

with equality iff all $\|b_i^*\|$ are equal.

Proof. The box has orthogonal edges of lengths $a_i = \|b_i^*\|$ and volume $\prod_i a_i = \det \Lambda$. Its second moment per dimension is $\frac{1}{n}\sum_i a_i^2/12$. Divide by $(\prod a_i)^{2/n}$ and apply the AM–GM inequality to $a_1^2, \ldots, a_n^2$. □

Remark C.10 (Babai's round-off algorithm). Babai's other algorithm (Babai, 1986), round-off, maps x to $B\lfloor B^{-1}x \rceil$, where the columns of B are the basis vectors. Its cells are translates of the parallelepiped $B[-\frac{1}{2}, \frac{1}{2})^n$, which has volume $|\det B|$ and second moment $\mathbb{E}\|Bc\|^2 = \operatorname{tr}(BB^\top)/12$ for $c \sim \text{Unif}([-\frac{1}{2}, \frac{1}{2})^n)$. Hence

$$G_{\text{round-off}} = \frac{\operatorname{tr}(BB^\top)}{12\,n\,|\det B|^{2/n}} \;\geq\; \frac{1}{12},$$

by the AM–GM inequality applied to the squared singular values of B, with equality iff all singular values are equal.

So neither variant of Babai's algorithm, with any basis of any lattice, beats scalar rounding. Realising $G_{E_8} < 1/12$ requires the exact decoder above.

D DATA-FREE ROPE-PAIR KEY SHAPING

Key shaping is part of the initial configuration v1 only; HybridQuant and HybridQuant-G do not use it, because it cost perplexity in the design ablation (Appendix B.1). Keys are scaled by a positive vector c before rotation and unscaled after dequantization. The model therefore computes $q \cdot k = (q \oslash c) \cdot (c \odot k)$ exactly; only the distribution of the quantization error changes. With $\alpha = 1$ and the notation of Algorithm 1, the logit error is

$$\delta = \frac{1}{\sqrt{d}}\, q \cdot \big(c^{-1} \odot Q_R^\top \Delta e\big), \qquad \Delta = \hat{\sigma} s, \quad \hat{\sigma}^2 \approx \tfrac{1}{d}\|c \odot k\|^2. \tag{2}$$

Lemma D.1 (Commutation with RoPE)**.** *RoPE rotates each coordinate pair (j, j') ($j' = j + d/2$ in the rotate-half convention) by a position-dependent angle. $\mathrm{diag}(c)$ commutes with every such rotation if and only if $c_j = c_{j'}$ for every pair.*

Proof. On a pair, $\mathrm{diag}(c_j, c_{j'})$ commutes with every 2×2 rotation iff it is a multiple of the identity. □

Proposition D.2 (Optimal shaping exponent)**.** *Assume the query and key are independent with second moments $\lambda_j = \mathbb{E}q_j^2$ and $a_j = \mathbb{E}k_j^2$. Then the dither-averaged logit error variance is*

$$\mathbb{E}\,\delta^2 = \frac{G_{E_8} s^2}{d^2}\, f(c), \qquad f(c) = \Big(\sum_j c_j^2 a_j\Big)\Big(\sum_j \frac{\lambda_j}{c_j^2}\Big),$$

and $f(c) \geq \big(\sum_j \sqrt{a_j \lambda_j}\big)^2$, with equality iff $c_j^4 \propto \lambda_j / a_j$. In particular:

- *(i) without data, i.e. with a_j equal, the optimum is $c_j \propto \lambda_j^{1/4}$ and the gain over no shaping is $f(\mathbf{1})/f(c^\star) = d \sum_j \lambda_j / (\sum_j \sqrt{\lambda_j})^2 \geq 1$;*
- *(ii) the exponent $1/2$, $c_j \propto \lambda_j^{1/2}$, gives $f = f(\mathbf{1})$ exactly when the a_j are equal, i.e.* no *gain.*

Proof. By Lemma C.2 and the orthogonality of Q_R, $Q_R^\top \Delta e$ has covariance $G_{E_8} \Delta^2 I_d$. Hence $\mathbb{E}_u \delta^2 = \frac{G_{E_8} \Delta^2}{d} \sum_j q_j^2 / c_j^2$ with $\Delta^2 = s^2 \|c \odot k\|^2 / d$; take expectations over the independent q and k. Cauchy–Schwarz applied to the vectors $(c_j \sqrt{a_j})_j$ and $(\sqrt{\lambda_j}/c_j)_j$ gives the bound and the equality condition. For (ii), with $c_j^2 = \lambda_j$ and $a_j \equiv a$, $f = a \sum_j \lambda_j \cdot d = f(\mathbf{1})$. □

The data-free estimate. Queries are $q = W_q\, \mathrm{RMSNorm}(h)$. Treating the normalised hidden state as isotropic gives $\lambda_j \propto \|(W_q)_{j,:} \odot \gamma\|^2$, where γ is the RMSNorm gain; for models with a query norm we use the square of its weight. We average over the query heads that share a KV head (GQA) and over the two coordinates of each RoPE pair, so that c commutes with RoPE (Lemma D.1). The pair average is also the position-averaged energy of the *post*-RoPE query, since $\mathbb{E}_\theta(\cos\theta\, q_j - \sin\theta\, q_{j'})^2 = \frac{1}{2}(\lambda_j + \lambda_{j'})$ when θ is uniform. Finally $c_j = p_j^{1/4}$ is normalised to geometric mean one per head; f is scale-invariant, so the normalisation does not change the objective. Only the checkpoint is used.

PROVED under the stated second-moment model. The equal-key-energy proxy is an assumption; the ablation `abl-noshape` measures the effect on real models.

E RANDOMIZED HADAMARD ROUNDS: WHAT THE LATTICE SEES

E.1 STRUCTURE

Lemma E.1 (Basic properties)**.** *$Q_R = H D_R \cdots H D_1$ is orthogonal for every $R \geq 0$ and every choice of signs. $Q_R^{-1} = Q_R^\top = D_1 H \cdots D_R H$ (Sylvester H is symmetric). Applying the same Q_R to queries and keys preserves every logit, $(Q_R q) \cdot (Q_R k) = q \cdot k$. Applying Q_R costs $R\, d \log_2 d$ additions/subtractions and Rd sign flips per row.*

Proof. It is a product of orthogonal matrices. The cost is that of R fast Walsh–Hadamard transforms. □

All rounds act on the *same* (head-dimension) axis. Mixing across heads is not allowed, and the next proposition shows why.

Proposition E.2 (Per-head logits force a block-diagonal transform)**.** *Let P_h be the coordinate projection onto head h in $\mathbb{R}^{Hd}$ and let $M \in O(Hd)$. If $(P_h M q) \cdot (P_h M k) = (P_h q) \cdot (P_h k)$ for all $q, k \in \mathbb{R}^{Hd}$ and all h, then M commutes with every P_h, i.e. M is block diagonal over heads.*

Proof. The hypothesis says $M^\top P_h M = P_h$ as bilinear forms. Multiply by M on the left: $P_h M = M P_h$. □

Proposition E.3 (Averaging rotations is not a rotation)**.** *Let $Q^{(1)}, \dots, Q^{(K)}$ be orthogonal and $\bar{Q} = \frac{1}{K}\sum_i Q^{(i)}$. Then $\|\bar{Q}x\| \le \|x\|$ for all x. If the $Q^{(i)}$ are not all equal, there is an x with $\|\bar{Q}x\| < \|x\|$, so $\bar{Q}$ is not orthogonal and quantizing $\bar{Q}x$ loses exact invertibility.*

Proof. $\|\bar{Q}x\| \le \frac{1}{K}\sum_i \|Q^{(i)}x\| = \|x\|$. The Euclidean norm is strictly convex, so equality forces $Q^{(1)}x = \cdots = Q^{(K)}x$. If this held for every x, all $Q^{(i)}$ would coincide. □

The anisotropy fallback therefore *selects* one of K sign sets per row (the one with the smallest μ); this keeps the transform orthogonal and costs $\log_2 K$ bits per row, which are counted in equation 1.

E.2 Exact and high-probability properties of one round

Condition on the previous round's output $z = y^{(r-1)}$ and write $y_i^{(r)} = \sum_j a_j \epsilon_{r,j}$ with $a_j = H_{ij} z_j$.

Lemma E.4 (Exactly flat variance)**.** *For every i and every z,* $\mathrm{Var}(y_i^{(r)} \mid z) = \sum_j a_j^2 = \|z\|^2/d = \sigma^2$.

Proof. $H_{ij}^2 = 1/d$. □

Lemma E.5 (Anisotropy after one round, any input)**.** *For every fixed $w \in \mathbb{R}^d$ and $\delta \in (0,1)$,*

$$\mathbb{P}_D\Big(\mu(HDw) > \frac{2\log(2d/\delta)}{d}\Big) \le \delta.$$

Proof. By Hoeffding's inequality (Hoeffding, 1963), $\mathbb{P}(|(HDw)_i| > t) \le 2\exp(-t^2 d/(2\|w\|^2))$. Take $t^2 = 2\|w\|^2 \log(2d/\delta)/d$, a union bound over the d coordinates, and use $\|HDw\| = \|w\|$. □

Lemma E.6 (Third moment after one round, any input)**.** *For every fixed w,* $\mathbb{E}_D\|HDw\|_3^3 \le 2\sqrt{2/\pi}\, d\, \sigma(w)^3 \approx 1.596\, d\sigma^3$.

Proof. Each coordinate is a Rademacher sum with variance σ^2. Khintchine's inequality with Haagerup's sharp constant for $p = 3$ (Haagerup, 1981), which equals the Gaussian moment $\mathbb{E}|g|^3 = 2\sqrt{2/\pi}$, gives $\mathbb{E}|(HDw)_i|^3 \le 2\sqrt{2/\pi}\,\sigma^3$. Sum over i. □

E.3 Berry–Esseen: one round depends on the input, two do not

Theorem E.7 (Shape of the rotated coordinates)**.** *Let $C_0 = 0.56$ (Shevtsova, 2010). For every $r \ge 1$, i and $z = y^{(r-1)}$,*

$$d_{\mathrm{K}}\Big(\mathcal{L}\big(y_i^{(r)}/\sigma \mid z\big), \Phi\Big) \le C_0 \frac{\|z\|_3^3}{\|z\|_2^3} \le C_0\sqrt{\mu(z)}. \tag{3}$$

Consequently:

(a) (One round depends on the input.) *For $R = 1$ the bound is $C_0\|x\|_3^3/\|x\|_2^3$, which can be of order one. For a spike $x = \sqrt{d}\,\sigma e_m$, $y_i^{(1)} = \pm\sigma$ exactly and $d_{\mathrm{K}} = \Phi(1) - \frac{1}{2} = 0.3413$ for every d. For k equal spikes (k even), $y_i^{(1)}/\sigma = k^{-1/2}\sum_{l=1}^{k}\eta_l$ with i.i.d. signs, and $d_{\mathrm{K}} \geq \frac{1}{2}\binom{k}{k/2}2^{-k} \asymp k^{-1/2}$, independently of d.*

(b) (Two or more rounds are universal.) *For every $R \geq 2$, every input x and every choice of $D_1, \ldots, D_{R-2}$, with probability at least $1-\delta$ over D_{R-1},*

$$d_{\mathrm{K}} \leq C_0\sqrt{2\log(2d/\delta)/d},$$

and averaged over D_{R-1},

$$\mathbb{E}_{D_{R-1}} d_{\mathrm{K}} \leq C_0 \cdot 2\sqrt{2/\pi}\, d^{-1/2} \leq 0.894\, d^{-1/2}.$$

Proof. equation 3 is the Berry–Esseen inequality for independent, non-identically distributed summands with constant $C_0 \leq 0.56$ (Shevtsova, 2010). Here $\sum_j \mathbb{E}|a_j\epsilon_j|^3 = d^{-3/2}\|z\|_3^3$ and $\sigma^3 = d^{-3/2}\|z\|_2^3$ (Lemma E.4). The second inequality is $\|z\|_3^3 \leq \|z\|_\infty\|z\|_2^2$. (a) For a spike, $y_i^{(1)} = \sqrt{d}\,\sigma H_{im}\epsilon_{1,m} = \pm\sigma$, a fair two-point law. Its CDF equals $\frac{1}{2}$ on $[-1,1)$, so $d_{\mathrm{K}} = \sup_{t\in[-1,1)}|\frac{1}{2} - \Phi(t)| = \Phi(1) - \frac{1}{2}$. For k equal spikes the sum has an atom of mass $\binom{k}{k/2}2^{-k}$ at 0. A law with an atom of mass J is at distance at least $J/2$ from any continuous CDF. (b) Apply equation 3 at round R with $z = y^{(R-1)} = HD_{R-1}w$, where $w = y^{(R-2)}$ is arbitrary. Lemma E.5 bounds $\mu(z)$ with probability $1-\delta$, and Lemma E.6 bounds $\mathbb{E}\|z\|_3^3/\|z\|_2^3$ because $\|z\|_2 = \|w\|_2$ is deterministic. □

The key point is *which* vector controls the bound. For $R = 1$ it is the input x, which the data choose. For $R \geq 2$ it is the output of a random round, which Lemma E.5 controls for every input. This is what makes a single, data-free code model valid for all rows.

Remark E.8 (Per-row versus per-coordinate statements). Theorem E.7 concerns the law of a single coordinate over the sign draws. The expected empirical CDF of the d coordinates of one row is the average of these marginals, so the same bounds hold for it.

F Peaks and coherence after R rounds

F.1 The peak $\|y\|_\infty$

Theorem F.1 (Peak after R rounds). *Let $x \neq 0$, $\sigma = \sigma(x)$ and $y = Q_R x$.*

(i) For every R, $\|y\|_\infty \geq \sigma$.

(ii) For every $R \geq 1$, $\mathbb{P}\big(\|y\|_\infty > \sigma\sqrt{2\log(2d/\delta)}\big) \leq \delta$.

(iii) For $R = 1$ and x supported on k coordinates, $\|y\|_\infty \leq \|x\|_1/\sqrt{d} \leq \sqrt{k}\,\sigma$ deterministically. For a spike ($k = 1$) this equals σ, the smallest possible value.

(iv) For every $R \geq 2$ and every x, with probability at least $1-\delta$ over D_{R-1},

$$\mathbb{E}_{D_R}\max_i y_i \;\geq\; \frac{\sigma}{L}\min\Big(\sqrt{\log\tfrac{d}{\beta+1}},\ \sqrt{\tfrac{d}{2\log(2d/\delta)}}\Big), \qquad \beta = 8\log(2d/\delta),$$

where L is a universal constant, and $\mathbb{P}_{D_R}\big(|\max_i y_i - \mathrm{Med}| \geq t\big) \leq 4e^{-t^2/(16\sigma^2)}$.

Proof. (i) The maximum is at least the RMS. (ii) Apply Lemma E.5 to the last round. (iii) $|(HDx)_i| \leq \sum_j |H_{ij}x_j| = \|x\|_1/\sqrt{d}$, and $\|x\|_1 \leq \sqrt{k}\|x\|_2$.

(iv) Condition on $z = y^{(R-1)}$ and write $y_i = \langle t_i, \epsilon_R\rangle$ with $t_i = (H_{ij}z_j)_j$. By Lemma E.5, $\mu(z) \leq 2\log(2d/\delta)/d$ with probability $1-\delta$. Let $p_j = z_j^2/\|z\|^2$. The correlations are $r_{il} = \langle t_i, t_l\rangle/\sigma^2 = \sum_j dH_{ij}H_{lj}p_j$. For Sylvester H, $dH_{ij}H_{lj} = \sqrt{d}\,H_{i\oplus l,j}$, so $r_{il} = \rho_{i\oplus l}$ with $\rho = \sqrt{d}\,Hp$. Parseval

gives $\sum_m \rho_m^2 = d\|p\|_2^2 \le d\max_j p_j = d\mu(z)$, and $\rho_0 = \sum_j p_j = 1$. Hence at most $4d\mu(z) \le \beta$ offsets $m \neq 0$ have $|\rho_m| > \frac{1}{2}$. Greedily choose $I \subseteq [d]$ such that $i \oplus l$ is never such an offset for distinct $i, l \in I$. Each choice excludes at most $\beta + 1$ indices, so $|I| \ge d/(\beta+1)$. For $i \neq l$ in I, $\|t_i - t_l\|^2 = 2\sigma^2(1 - r_{il}) \ge \sigma^2$. Moreover $\|t_i\|_\infty = \|z\|_\infty/\sqrt{d} = \sqrt{\mu(z)}\,\sigma$. Sudakov's minoration for Bernoulli processes (Talagrand, 2021, Ch. 6) states that points with pairwise distances at least a and $\|t_i\|_\infty \le b$ satisfy $\mathbb{E}\max_i\langle t_i, \epsilon\rangle \ge L^{-1}\min(a\sqrt{\log|I|}, a^2/b)$. Apply it with $a = \sigma$ and $b = \sqrt{\mu(z)}\,\sigma$. Finally, $\epsilon \mapsto \max_i\langle t_i, \epsilon\rangle$ is convex and σ-Lipschitz, since $\max_i \|t_i\|_2 = \sigma$. Talagrand's convex-distance inequality (Talagrand, 1995), applied to $(\epsilon+1)/2 \in [0,1]^d$, gives the tail bound. □

Theorem F.2 (Three rounds do not reduce the peak of sparse outliers)**.** *If x is supported on k coordinates, then $\|Q_1 x\|_\infty \le \sqrt{k}\,\sigma$, while for every $R \ge 2$, with probability $1-\delta$ over D_{R-1}, $\mathbb{E}_{D_R}\|Q_R x\|_\infty \ge \frac{\sigma}{L}\min(\cdots)$ as in Theorem F.1(iv). Hence for $k < L^{-2}\min\big(\log\frac{d}{\beta+1}, \frac{d}{2\log(2d/\delta)}\big)$, one round has a strictly smaller peak than two or three.*

Proof. Combine Theorem F.1(iii) and (iv), using $\|y\|_\infty \ge \max_i y_i$. □

Proposition F.3 (Generic chaining cannot separate $R = 1$ from $R = 3$)**.** *Fix a set X of inputs and a coordinate l. Conditional on $D_1, \dots, D_{R-1}$, the process $x \mapsto (Q_R x)_l$ is a Rademacher process in ϵ_R whose canonical sub-Gaussian pseudometric is $\|Q_{R-1}(x - x')\|/\sqrt{d} = \|x - x'\|/\sqrt{d}$, the same for every $R \ge 1$. Hence every bound on $\mathbb{E}\sup_{x\in X}\|Q_R x\|_\infty$ obtained from Talagrand's γ_2 functional or Dudley's integral (Dudley, 1967) is identical for $R = 1$ and $R = 3$.*

Proof. The increments are $\sum_j H_{lj}\epsilon_{R,j}(Q_{R-1}(x - x'))_j$, which are sub-Gaussian with variance proxy $\|Q_{R-1}(x - x')\|^2/d$ by Hoeffding. Q_{R-1} is orthogonal. □

A separation between the numbers of rounds must therefore come from finer structure than γ_2, such as the ℓ_1 part of the Bednorz–Latała decomposition (Bednorz & Latała, 2014), which is what drives Theorem F.2, or the lattice structure in Appendix G.

F.2 Coherence of a systematic outlier subspace

LLM outliers are systematic: the same few channels, or a low-dimensional subspace, are large for almost every token (Dettmers et al., 2022; Sun et al., 2024). With a single sign draw shared by all tokens (needed for reproducibility), guarantees must hold *uniformly* over the subspace. Let $U \in \mathbb{R}^{d\times k}$ be an orthonormal basis of the outlier subspace $\mathcal{U}$ and write every token as $x_t = b_t + Uc_t$.

Definition F.4 (Coordinate and block coherence)**.** For an orthogonal Q let

$$\eta(Q) = \sup_{u\in\mathcal{U},\|u\|=1}\|Qu\|_\infty^2 = \max_l \|U^\top Q^\top e_l\|^2, \qquad \kappa(Q) = \frac{d}{8}\max_b \|P_b Q U\|_{\mathrm{op}}^2,$$

where P_b selects the eight coordinates of block b. The outlier part of *every* token puts at most $\eta\|Uc_t\|^2$ energy on any coordinate and at most $\frac{8}{d}\kappa\|Uc_t\|^2$ on any block. $\kappa = 1$ means "no more than the fair share".

Lemma F.5 (Lower bounds)**.** *For every orthogonal Q: $\eta(Q) \ge k/d$, $\max(1, k/8) \le \kappa(Q) \le d\,\eta(Q)$, and $\kappa(I) = d/8$ for coordinate-aligned $\mathcal{U}$.*

Proof. $\sum_l \|U^\top Q^\top e_l\|^2 = \|QU\|_F^2 = k$, so the maximum is at least k/d. Similarly $\sum_b \|P_b QU\|_F^2 = k$, so some block has $\|P_b QU\|_F^2 \ge 8k/d$, and $\|M\|_{\mathrm{op}}^2 \ge \|M\|_F^2/\min(8,k)$ for an $8 \times k$ matrix M. The upper bound holds because a block has eight coordinates. Without rotation a single coordinate outlier lies in one block. □

Lemma F.6 (One round is optimal for coordinate coherence)**.** *For every $\mathcal{U}$ and δ, with probability $1-\delta$ over D,*

$$\eta(HD) \le \frac{\big(\sqrt{2k} + 4\sqrt{\log(2d/\delta)}\big)^2}{d}.$$

If $\mathcal{U}$ is spanned by k coordinate vectors then $\eta(HD) = k/d$ exactly, for every D.

Proof. $\sup_u (HDu)_l = \|U^\top D h_l\| =: f_l(\epsilon)$, where h_l is row l of H. f_l is convex in ϵ and Lipschitz with constant $\|U^\top \operatorname{diag}(h_l)\|_{\mathrm{op}} \le \|h_l\|_\infty = d^{-1/2}$. Also $\mathbb{E} f_l^2 = \sum_j H_{lj}^2 \|U_{j,:}\|^2 = k/d$, so its median is at most $\sqrt{2k/d}$. Talagrand's convex-distance inequality for $(\epsilon+1)/2 \in [0,1]^d$ gives $\mathbb{P}(f_l \ge \sqrt{2k/d} + t) \le 2e^{-t^2 d/16}$; take a union bound over l. In the coordinate case, $\|U^\top D h_l\|^2 = \sum_{j \in S} H_{lj}^2 = k/d$. □

Dudley's entropy integral over the unit sphere of $\mathcal{U}$ ($\gamma_2 \asymp \sqrt{k}$) gives the same order with unspecified constants. Lemmas F.5 and F.6 show that for coordinate coherence one round is already optimal, and further rounds cannot improve it.

Lemma F.7 (One Sylvester round has exact block collisions)**.** *Write the row index as $i = 8b + a$ with $a \in \{0, \dots, 7\}$ and the column index as $j = 8 j_{\mathrm{hi}} + j_{\mathrm{lo}}$. Then*

$$H_{8b+a,\, j} = d^{-1/2} (-1)^{\langle b, j_{\mathrm{hi}} \rangle} (w_{j_{\mathrm{lo}}})_a, \qquad (w_m)_a = (-1)^{\langle a, m \rangle},$$

so the restriction of column j to any block is $\pm d^{-1/2} w_{j \bmod 8}$, one of eight orthogonal Walsh vectors. Consequently, for $\mathcal{U}$ spanned by coordinates S, every block b and every sign vector D,

$$\kappa(HD) = \max_{r \in \{0,\dots,7\}} \big| S \cap (r + 8\mathbb{Z}) \big|.$$

In the rotate-half RoPE convention the two coordinates of a pair, j and $j + d/2$, have the same residue mod 8 (for $d \ge 16$). A rotating RoPE outlier pair therefore has $\kappa(HD) = 2$ although $\max(1, k/8) = 1$.

Proof. The bitwise inner product splits over bit positions: $\langle 8b + a, 8 j_{\mathrm{hi}} + j_{\mathrm{lo}} \rangle = \langle b, j_{\mathrm{hi}} \rangle + \langle a, j_{\mathrm{lo}} \rangle$. Let $n_r = |S \cap (r + 8\mathbb{Z})|$. Then $M = P_b H D U$ has columns $\pm d^{-1/2} w_{j \bmod 8}$ and $M M^\top = d^{-1} \sum_r n_r w_r w_r^\top$. The w_r are orthogonal with $\|w_r\|^2 = 8$, so the eigenvalues of $M M^\top$ are $8 n_r / d$. □

The collision is deterministic: no choice of signs removes it, because the supremum over $u \in \mathcal{U}$ absorbs the signs. It is a structural property of a single structured round, not a low-probability event.

Lemma F.8 (A second round removes collisions up to a log factor)**.** *Let $V = QU$ for an orthogonal Q (e.g. $Q = Q_1$) and let $\eta_V = \max_l \|V_{l,:}\|^2$. With probability $1 - \delta$ over D,*

$$\kappa(HDQ) \le 2 \max\Big(1, \frac{d\,\eta_V}{8}\Big) \log \frac{(8+k)d}{8\delta}.$$

With $Q = Q_1$ and Lemma F.6, $\kappa(Q_2) \lesssim \log \frac{dk}{\delta} \cdot \max\big(1, \frac{(\sqrt{2k} + 4\sqrt{\log(2d/\delta)})^2}{8}\big)$, which matches the lower bound $\max(1, k/8)$ of Lemma F.5 up to logarithmic factors. The same bound holds for $\kappa(Q_3)$; the analysis does not separate $R = 2$ from $R = 3$ in κ.

Proof. $M_b = P_b H D V = \sum_l \epsilon_l B_l$ with $B_l = h_{b,l} V_{l,:}$, where $h_{b,l} \in \mathbb{R}^8$ is column l of H restricted to block b. Then $\sum_l B_l B_l^\top = \sum_l \|V_{l,:}\|^2 h_{b,l} h_{b,l}^\top \preceq \eta_V H_b H_b^\top = \eta_V I_8$ and $\sum_l B_l^\top B_l = \frac{8}{d} V^\top V = \frac{8}{d} I_k$. The matrix Rademacher tail bound (Tropp, 2012, Thm. 4.1.1) gives $\mathbb{P}(\|M_b\| \ge t) \le (8 + k) e^{-t^2/(2v)}$ with $v = \max(\eta_V, 8/d)$. Take a union bound over the $d/8$ blocks. □

G TWO VERSUS THREE ROUNDS: THE LATTICE FLOOR OF SPARSE OUTLIERS

The Berry–Esseen bound of Theorem E.7(b) is the same for every $R \ge 2$. We now show that it is tight for $R = 2$ on sparse outliers but not for $R = 3$. The reason is arithmetic: after two rounds the coordinates of a spike lie on a coarse lattice, and a third round refines it.

Throughout this section $d = 2^m \ge 16$, $\sigma = 1$, and the input is a spike $x = \sqrt{d}\, e_m$, which is the extreme case of Theorem E.7(a).

G.1 Two rounds: a floor of order $d^{-1/2}$

Proposition G.1 (Two-round floor). $y_i^{(2)} = (2B - d)/\sqrt{d}$ *with* $B \sim \mathrm{Bin}(d, \frac{1}{2})$. *Hence*

$$\frac{1}{2}\binom{d}{d/2}2^{-d} \;\le\; d_{\mathrm{K}}\big(y_i^{(2)}, \Phi\big) \;\le\; \frac{C_0}{\sqrt{d}}, \qquad \frac{1}{2}\binom{d}{d/2}2^{-d} \ge \frac{1}{2\sqrt{2d}}.$$

At $d = 128$ *the two bounds are* 0.0352 *and* 0.0495*; the exact value is* 0.0352 *(Table 16).*

Proof. $y_j^{(1)} = s_j := \sqrt{d}H_{jm}\epsilon_{1,m} \in \{\pm 1\}$, and $y_i^{(2)} = \sum_j H_{ij}\epsilon_{2,j}s_j = d^{-1/2}\sum_j \eta_j$, where $\eta_j = \sqrt{d}H_{ij}s_j\epsilon_{2,j}$ are i.i.d. Rademacher. The atom at 0 has mass $\binom{d}{d/2}2^{-d}$, and an atom of mass J forces $d_{\mathrm{K}} \ge J/2$. The upper bound is equation 3 with a flat $z = y^{(1)}$, for which $\|z\|_3^3/\|z\|_2^3 = d^{-1/2}$. The central binomial bound $\binom{2n}{n}4^{-n} \ge (4n)^{-1/2}$ is standard. □

G.2 Three rounds: order d^{-1} up to a logarithm

With $W = \sqrt{d}H$ and $\zeta_k = W_{ik}\epsilon_{3,k}$,

$$Z := y_i^{(3)} = \frac{1}{d}\,\zeta^\top W\eta = \sum_k \zeta_k c_k, \qquad c_k = \frac{N_k}{d}, \quad N = W\eta \in \mathbb{Z}^d, \tag{4}$$

where ζ and η are independent Rademacher vectors. Z is a decoupled Rademacher chaos of order two with matrix W/d, which has Hilbert–Schmidt norm 1 and operator norm $d^{-1/2}$. Conditional on η, Z is a weighted Rademacher sum with $\sum_k c_k^2 = \|W\eta\|^2/d^2 = 1$ *exactly*. Write $\varphi_\eta(u) = \prod_k \cos(uc_k)$ for its conditional characteristic function.

Theorem G.2 (Three rounds). *There are absolute constants* C *and* d_0 *such that for* $d = 2^m \ge d_0$

$$\frac{3}{8(d+1)} \;\le\; d_{\mathrm{K}}\big(y_i^{(3)}, \Phi\big) \;\le\; C\,\frac{\log d}{d}.$$

Hence, by Proposition G.1, for all sufficiently large d *the three-round distance is strictly smaller than the two-round distance, by a factor of order* $\sqrt{d}/\log d$.

The proof uses five lemmas. Lemmas G.3–G.5 are proved completely. Lemmas G.6 and G.7 rest on the local limit theorem for the binomial distribution; we give the argument and certify the resulting constants numerically by exact computation for $d \le 512$ (Table 17). The lower bound of Theorem G.2 is therefore proved, and the upper bound, which uses Lemmas G.6 and G.7, is a proof sketch.

Lemma G.3 (Common residue; lower bound). *All* N_k *have the same residue* $r \in \{0, 2\}$ *modulo* 4. *Consequently* $Z \in \frac{4}{d}\mathbb{Z}$ *and* $d_{\mathrm{K}}(Z, \Phi) \ge 3/(8(d+1))$.

Proof. For $k \ne l$, $N_k - N_l = 2\sum_{j:W_{kj}\ne W_{lj}} W_{kj}\eta_j$. Distinct Sylvester rows differ in exactly $d/2$ positions, so this sum of $d/2$ signs is even. Each $N_k = \sum_j W_{kj}\eta_j$ is even because d is even. Then $\sum_k \zeta_k N_k \equiv r\sum_k \zeta_k \equiv 0 \pmod 4$, because r is even and $\sum_k \zeta_k \equiv d \equiv 0 \pmod 2$. For the lower bound, $\mathbb{E}Z^2 = 1$, so $\mathbb{P}(|Z| \le 2) \ge 3/4$ by Chebyshev. $[-2, 2]$ contains $d+1$ points of $\frac{4}{d}\mathbb{Z}$, so some atom has mass at least $3/(4(d+1))$. □

Lemma G.4 (Fourth moment of the Walsh spectrum). $\sum_k N_k^4 = dQ$ *with* $Q = \sum_{a\oplus b\oplus c\oplus e=0} \eta_a\eta_b\eta_c\eta_e$ *(sum over ordered triples* (a, b, c)*,* $e = a \oplus b \oplus c$*). Moreover* $\mathbb{E}\sum_k N_k^4 = 3d^3 - 2d^2$, $\mathrm{Var}\sum_k N_k^4 \le 24d^5$, *and*

$$\mathbb{P}\Big(\sum_k N_k^4 > 6d^3\Big) \le \frac{8}{3d}.$$

Proof. $\sum_k W_{ka}W_{kb}W_{kc}W_{ke} = d\,\mathbf{1}[a \oplus b \oplus c \oplus e = 0]$ for Sylvester W. A term of Q equals 1 iff its indices pair up. There are exactly $3d^2 - 2d$ such ordered triples, by inclusion–exclusion over the three pairings. Every other term has four distinct indices. Each such 4-set arises from $4! = 24$ ordered triples, and products over distinct 4-sets are orthonormal. Hence $\mathrm{Var}Q = 24^2 \cdot \#\{\text{4-sets}\} \le 24d^3$. Chebyshev with deviation $3d^3$ gives the bound. (Hypercontractivity for order-four chaos, e.g. via Latała (2006), would improve the probability to $e^{-cd^{1/4}}$ but does not change the final rate.) □

Lemma G.5 (Central frequencies). *Let* $\mathcal{G} = \{\sum_k N_k^4 \le 6d^3\} \cap \{\|N\|_\infty \le \sqrt{2d\log(4d^2)}\}$. *Then* $\mathbb{P}(\mathcal{G}^c) \le 3.2/d$, *and on* $\mathcal{G}$:

(a) for $|u| \le d/\|N\|_\infty$: $|\varphi_\eta(u) - e^{-u^2/2}| \le 0.70\, u^4 e^{-u^2/2}/d$;

(b) for $|u| \le 0.4534\sqrt{d}$: $|\varphi_\eta(u)| \le e^{-u^2/4}$.

Proof. $\mathbb{P}(\mathcal{G}^c) \le \frac{8}{3d} + \frac{1}{2d}$ by Lemma G.4 and Hoeffding with a union bound. (a) Here $|uc_k| \le 1$ for all k. On $[-1,1]$, $-\frac{x^2}{2} - 0.116x^4 \le \log\cos x \le -\frac{x^2}{2}$, because the Taylor coefficients of $\log\cos x + x^2/2$ are all negative and sum to $\log\cos 1 + \frac{1}{2} = -0.1156$. Summing over k and using $\sum c_k^2 = 1$ and $\sum c_k^4 = \sum N_k^4/d^4 \le 6/d$ gives $\varphi_\eta = e^{-u^2/2}e^{-E}$ with $0 \le E \le 0.70u^4/d$. (b) Put $\theta = u/d$ and $M = \pi/(2\theta) \ge \sqrt{12d}$. Coordinates with $|N_k| \le M$ have $|\theta N_k| \le \pi/2$. The others carry energy $\sum_{|N_k|>M} N_k^2 \le \sum N_k^4/M^2 \le d^2/2$. Since $|\cos x| \le \exp(-\operatorname{dist}(x,\pi\mathbb{Z})^2/2)$, $|\varphi_\eta(u)| \le \exp(-\frac{\theta^2}{2}\sum_{|N_k|\le M} N_k^2) \le e^{-u^2/4}$. □

Lemma G.6 (A positive fraction of far coordinates). *Let* $p_r(\theta) = \mathbb{P}\big(\operatorname{dist}(\theta N_k, \pi\mathbb{Z}) \ge \pi/4 \mid N_k \equiv r \bmod 4\big)$. *There is an absolute* $p_0 > 0$ *with* $p_r(\theta) \ge p_0$ *for* $r \in \{0,2\}$ *and all* $\theta \in [0.4534/\sqrt{d}, \pi/8]$.

Proof sketch. Write $N_k = r + 4q$. The points θN_k advance in steps of $4\theta \le \pi/2$ around a circle of length π. The far arc has length $\pi/2$, at least one step, so no step can jump over it. *Large* $\theta\sqrt{d}$: the law of q is a binomial restricted to a residue class, which is unimodal. Combining this with the no-jump property gives $p_r \ge \frac{1}{4}$ once $\theta\sqrt{d}$ exceeds an absolute constant K. $\theta\sqrt{d} \in [0.4534, K]$: by the local limit theorem (Petrov, 1975), p_r converges uniformly to a wrapped-Gaussian probability that is continuous and positive on this compact range. Exact computation over a 4000-point grid gives $\min_{r,\theta} p_r(\theta) = 0.0769, 0.0755, 0.0793, 0.0767$ for $d = 64, 128, 256, 512$ (Table 17). □

Lemma G.7 (Pairwise near-independence). *For* $k \ne l$, $(N_k, N_l) \overset{d}{=} (S_1 + S_2,\, S_1 - S_2)$ *with* S_1, S_2 *independent sums of* $d/2$ *Rademacher variables; the joint law is the same for all pairs. Let* $C_\theta = \#\{k : \operatorname{dist}(\theta N_k, \pi\mathbb{Z}) \ge \pi/4\}$. *There is an absolute* c_1 *with* $|\operatorname{Cov}(\mathbf{1}_k, \mathbf{1}_l \mid r)| \le c_1/d$, *hence*

$$\operatorname{Var}(C_\theta \mid r) \le \big(\tfrac{1}{4} + c_1\big)d, \qquad \mathbb{P}\big(C_\theta < \tfrac{dp_0}{2} \,\big|\, r\big) \le \frac{4(1/4 + c_1)}{p_0^2 d}.$$

Proof sketch. Distinct Sylvester rows agree on exactly $d/2$ coordinates, which gives the representation. (S_1, S_2) has an exact product law. Given r, the image $(S_1 + S_2, S_1 - S_2)$ ranges over the product lattice $(r + 4\mathbb{Z})^2$. A 45° rotation of a product of discrete Gaussians is again a product of discrete Gaussians, so by the local limit theorem with its first Edgeworth correction the joint law is within total variation $O(1/d)$ of the product of its marginals. The skewness term vanishes by symmetry. Exact computation gives $\max_{r,\theta} d\,|\operatorname{Cov}| = 0.121, 0.117, 0.113, 0.113$ for $d = 64$ to 512 (Table 17). Chebyshev gives the tail bound. □

Proof of Theorem G.2. The lower bound is Lemma G.3. For the upper bound we use Esseen's smoothing inequality (Feller, 1971, XVI.3) with $T = \pi d/8$, which lies below the conditional period $\pi d/4$ of $|\varphi_\eta|$ (Lemma G.3):

$$d_{\mathrm{K}}(Z, \Phi) \le \frac{2}{\pi}\int_0^T \frac{\mathbb{E}_\eta|\varphi_\eta(u) - e^{-u^2/2}|}{u}\, du + \frac{24}{\pi\sqrt{2\pi}\, T}.$$

The last term is $7.76/d$. Split the integral as follows.

1. On $\mathcal{G}$ and $u \le d/\|N\|_\infty$: by Lemma G.5(a) the contribution is at most $\frac{2}{\pi}\cdot\frac{0.70}{d}\int_0^\infty u^3 e^{-u^2/2}du = \frac{0.89}{d}$.
2. On $\mathcal{G}$ and $d/\|N\|_\infty \le u \le 0.4534\sqrt{d}$: by Lemma G.5(b) the integrand is at most $2e^{-u^2/4}/u$. Since $d/\|N\|_\infty \ge \sqrt{d/(2\log 4d^2)}$ on $\mathcal{G}$, the contribution is $e^{-\Omega(d/\log d)}$.

3. On $\mathcal{G}^c$ and $u \le 0.4534\sqrt{d}$: both characteristic functions have mean 0 and variance 1, so the integrand is at most $\min(u^2, 2)/u$. The contribution is at most $\frac{2}{\pi} \cdot \frac{3.2}{d}(2 + \log d) = O(\log d/d)$.

4. $0.4534\sqrt{d} \le u \le T$, i.e. $\theta \in [0.4534/\sqrt{d}, \pi/8]$: each far coordinate contributes $\cos^2 \le e^{-\pi^2/16}$. By Lemmas G.6 and G.7, $\mathbb{E}_\eta|\varphi_\eta(u)| \le \frac{4(1/4+c_1)}{p_0^2 d} + e^{-\pi^2 p_0 d/64}$, and $e^{-u^2/2} \le e^{-0.1d}$. Integrating du/u over this range contributes $O(1/d) \cdot \log \frac{T}{0.4534\sqrt{d}} = O(\log d/d)$.

Summing the four contributions and the smoothing term proves the theorem. □

Remark G.8 (Tightness and the logarithm). Exact computation (Table 16) gives $d \cdot d_{\mathrm{K}} = 1.17$–$1.28$ for $d = 32, \dots, 1024$ with no visible logarithm, while $\sqrt{d} \cdot d_{\mathrm{K}} \to 1/\sqrt{2\pi} = 0.399$ for two rounds. The $\log d$ in the theorem comes from the Chebyshev step in Lemma G.7; we conjecture the sharp rate is $\Theta(1/d)$.

Remark G.9 (Scope). Proposition G.1 and Theorem G.2 are proved for a spike, the extreme case of Theorem E.7(a); k equal spikes behave similarly (Table 15). For a general input the three-round output is a trilinear Rademacher chaos. We do not have a proof for that case beyond the $R \ge 2$ bound of Theorem E.7(b).

Remark G.10 (Where Latała's estimates enter). Two-sided moment bounds for decoupled chaos (Latała, 2006) give $\|Z\|_p \asymp \sqrt{p} + p/\sqrt{d}$, i.e. Gaussian tails up to $|t| \asymp \sqrt{d}$. Tail equivalence cannot detect lattice atoms, and the two-round binomial also has Gaussian tails. The Kolmogorov rate needs the arithmetic step (Lemmas G.6 and G.7); chaos moment bounds enter only through Lemma G.4.

H Where anisotropy enters the pipeline, and where it does not

Lemma H.1 (No overload)**.** *For every row, $|y_i| \le \|y\|_2 = \sqrt{d}\,\sigma$. Hence every dithered coordinate satisfies $|y_i/\Delta + u_i| \le \sqrt{d}/(s\,(1 - 2^{-11})) + 1$, and the doubled codes stay within int16 whenever $\sqrt{d}/s \le 16000$. For $d = 128$ this holds for every design SNR up to 74 dB. Rows are therefore never clipped; large coordinates are entropy-coded, with escape symbols, and cost only rate.*

Proof. $|y_i| \le \|y\|_2$. The nearest E_8 point lies within the covering radius 1 in Euclidean norm, hence within 1 in every coordinate. A doubled code is therefore at most $2(\sqrt{d}/(s(1-2^{-11}))+2) \le 32767$ when $\sqrt{d}/s \le 16000$. With $s = (G_{E_8} 10^{\mathrm{SNR}/10})^{-1/2}$ this is $\mathrm{SNR} \le 10\log_{10}(16000^2/(dG_{E_8}))$. □

Theorem H.2 (Distortion does not depend on anisotropy)**.** *Consider Algorithm 1 with Voronoi dither and $\alpha = 1$, any number of rounds $R \ge 0$, and any input rows (spiky, heavy-tailed, correlated). For every row, $\hat{y} - y = \Delta e$ with e uniform on $\mathcal{V}_{E_8}^{d/8}$ and independent of the data. In particular, the per-row SNR equals the design value (Proposition C.5), the logit error of equation 2 has variance $\frac{G_{E_8}\Delta^2}{d}\|q \oslash c\|^2$, and neither depends on R or on $\mu(x)$.*

Proof. Lemma H.1 rules out clipping, so Theorem C.4 applies exactly, blockwise and for any input. The logit variance follows from Lemma C.2 and the orthogonality of Q_R. □

Of the modes of Appendix B.1, only HybridQuant-G satisfies these hypotheses: the default HybridQuant has no dither, and v1 has $\alpha < 1$. Theorem H.2 is verified in Table 13 (SNR 11.71–11.72 dB against a 12 dB design for every $R \in \{0, 1, 2, 3\}$ and every outlier pattern; normalised logit error 0.98–1.04). It is also a unit test in the released code (`test_dithered_distortion_is_independent_of_rounds`).

Where anisotropy does enter.

1. *Rate.* The static code model is designed for a Gaussian source. A row with a large peak produces symbols beyond the model's alphabet (escapes, 16 raw bits each). The peak of every token's outlier part is controlled uniformly by η (Definition F.4), which one round already makes optimal (Lemma F.6). Measured: escape rate 0.73–0.77% at $R = 0$ and exactly 0 for $R \ge 1$ (Table 18).

2. *Rate predictability.* With $R \geq 1$ the measured payload matches the design rate (2.17–2.18 vs. 2.183 bits); with $R = 0$ it deviates (1.86–2.14) (Table 13). Deviations in either direction break the data-free rate targeting. On real models the effective rate is within 0.003 bits of the target in every run with $R \geq 1$ and deviates by up to 0.012 bits with $R = 0$ (Appendix B.2).
3. *Undithered variants*, including the default HybridQuant. Without dither Theorem C.4 does not apply, and the error depends on the input. Measured: SNR $16.2 \to 14.4 \to 11.9 \to 11.7$ dB for $R = 0, 1, 2, 3$ against the 11.72 dB design (Table 14); only $R \geq 2$ brings it to the design value.
4. *Fixed-rate absmax quantizers* (used by rotation-based baselines) are where anisotropy enters the distortion directly (Proposition H.3).

Proposition H.3 (Fixed-rate absmax quantization pays for the peak)**.** *A b-bit uniform quantizer on $[-\|y\|_\infty, \|y\|_\infty]$ with step $\Delta = 2\|y\|_\infty/(2^b - 1)$ and a uniform error (exact with subtractive dither, approximate at high resolution otherwise) has*

$$\frac{\mathbb{E}\|\hat{y} - y\|^2}{\|y\|^2} = \frac{d\,\Delta^2/12}{\|y\|^2} = \frac{d\,\mu(y)}{3(2^b - 1)^2}.$$

Proof. Substitute Δ and $\mu(y) = \|y\|_\infty^2/\|y\|^2$. □

Here the relative error is proportional to $d\mu(y)$. For such quantizers the rotation reduces the distortion itself. For sparse outliers Theorem F.2 shows that one round gives a smaller peak than more rounds. This matches Table 13: absmax 2-bit SNR is 9.7 dB at $R = 1$ against 6.1–6.9 dB at $R = 2, 3$ for a RoPE outlier pair. Our quantizer is designed so that this mechanism does not apply (Theorem H.2).

I Attention-level guarantees that do not degrade with context length

Fix a query q and a context of T keys and values. After quantization (Algorithm 1, $\alpha = 1$), with rotated-domain errors e_t^K, e_t^V,

$$\hat{k}_t = k_t + c^{-1} \odot Q_R^\top e_t^K, \qquad \hat{v}_t = v_t + Q_R^\top e_t^V, \qquad \hat{\ell}_t = \ell_t + \delta_t, \quad \delta_t = \frac{\tilde{q} \cdot e_t^K}{\sqrt{d}}, \;\; \tilde{q} = Q_R(q \oslash c),$$

and the attention output is $\hat{o} = \sum_t \hat{a}_t \hat{v}_t$ with $\hat{a} = \mathrm{softmax}(\hat{\ell})$. The exact output is $o = \sum_t a_t v_t$. Every block of e_t^K (resp. e_t^V) lies in the scaled cell $\Delta_t^K \mathcal{V}_{E_8}$ (resp. $\Delta_t^V \mathcal{V}_{E_8}$). Its diameter is 2Δ because the covering radius is 1. Let $\bar{\Delta} = \max_t \max(\Delta_t^K, \Delta_t^V)$ and $V_{\max} = \max_t \|v_t\| + \sqrt{d/8}\, \max_t \Delta_t^V \geq \max_t \|\hat{v}_t\|$.

Lemma I.1 (Softmax is a total-variation contraction)**.** *For all $\ell, \ell' \in \mathbb{R}^T$, $\|\mathrm{softmax}(\ell') - \mathrm{softmax}(\ell)\|_1 \leq 2\|\ell' - \ell\|_\infty$.*

Proof. The Jacobian is $J_{ij} = a_i(\mathbf{1}[i = j] - a_j)$. For any δ, $\|J\delta\|_1 = \sum_i a_i|\delta_i - \sum_j a_j \delta_j| \leq 2\sum_i a_i|\delta_i| \leq 2\|\delta\|_\infty$, because $\sum_i a_i = 1$ at *every* ℓ. Integrate along the segment from ℓ to ℓ'. □

Lemma I.2 (A Lipschitz constant independent of T)**.** *As a function of the concatenated errors $(e^K, e^V) \in \mathbb{R}^{2Td}$ with the Euclidean norm, $\hat{o}$ is L-Lipschitz with*

$$L = \sqrt{L_K^2 + L_V^2}, \qquad L_K = \frac{2V_{\max}\|q \oslash c\|}{\sqrt{d}}, \qquad L_V = 1.$$

Proof. For fixed e^K, $\hat{o}$ is affine in e^V with linear part $(e_t^V)_t \mapsto \sum_t a_t Q_R^\top e_t^V$. Its operator norm is $\|a\|_2 \leq \sqrt{\sum_t a_t} = 1$. For fixed e^V, $\|\hat{o}(e^K) - \hat{o}(e'^K)\| \leq V_{\max}\|\hat{a} - \hat{a}'\|_1 \leq 2V_{\max}\|\delta - \delta'\|_\infty$ by Lemma I.1. Moreover $|\delta_t - \delta_t'| \leq \|\tilde{q}\|\,\|e_t^K - e_t'^K\|/\sqrt{d} \leq \|q \oslash c\|\,\|e^K - e'^K\|/\sqrt{d}$. Combine the two directions with the triangle inequality and Cauchy–Schwarz. □

Using $\|\cdot\|_\infty$ in Lemma I.1 is what removes T. A per-key bounded-differences (McDiarmid) argument instead sums T worst-case contributions and gives a variance proxy that grows linearly in T.

Proposition I.3 (Deterministic worst-case error)**.** *For every dither realisation,*

$$\|\hat{o} - o\| \;\le\; \frac{2V_{\max}\|q \oslash c\|}{\sqrt{8}} \max_t \Delta_t^K + \sqrt{\frac{d}{8}} \max_t \Delta_t^V,$$

which does not depend on T.

Proof. $\|e_t^K\| \le \sqrt{d/8}\,\Delta_t^K$ (each of the $d/8$ blocks is within Δ_t^K of the origin), so $\|\delta\|_\infty \le \|q \oslash c\| \max_t \Delta_t^K/\sqrt{8}$. Then $\|\sum_t(\hat{a}_t - a_t)\hat{v}_t\| \le V_{\max}\|\hat{a} - a\|_1$ by Lemma I.1, and $\|\sum_t a_t Q_R^\top e_t^V\| \le \max_t \|e_t^V\|$. □

Theorem I.4 (Concentration of the attention output)**.** *Assume independent dithers for all (position, block) pairs. Then for every unit vector w,*

$$\mathrm{Var}\big(\langle w, \hat{o}\rangle\big) \le \frac{4\bar{\Delta}^2}{\pi^2}L^2, \qquad \mathbb{E}\|\hat{o} - \mathbb{E}\hat{o}\|^2 \le \frac{4d\,\bar{\Delta}^2}{\pi^2}L^2,$$

and there are universal constants c, C with $\mathbb{P}\big(|\langle w, \hat{o}\rangle - \mathbb{E}\langle w, \hat{o}\rangle| \ge t\big) \le C\exp\big(-c\,t\pi/(2\bar{\Delta}L)\big)$. None of these bounds depends on T. Since $\bar{\Delta} \propto s \propto 10^{-\mathrm{SNR}/20}$, they decay exponentially in the bit-rate.

Proof. By Theorem C.4, the $2T \cdot d/8$ block errors are independent, and block b is uniform on a convex body of diameter $2\Delta_b \le 2\bar{\Delta}$. The uniform law on a convex body of diameter D satisfies the Poincaré inequality $\mathrm{Var} f \le (D^2/\pi^2)\mathbb{E}\|\nabla f\|^2$ (Payne & Weinberger, 1960; Bebendorf, 2003). For product measures the Poincaré constant is the maximum of the factors' constants (by the Efron–Stein inequality, $\mathrm{Var} f \le \sum_b \mathbb{E}\mathrm{Var}_b f$). So $\mathrm{Var} f \le (4\bar{\Delta}^2/\pi^2)\mathbb{E}\|\nabla f\|^2$ for every locally Lipschitz f. Apply this to $f = \langle w, \hat{o}\rangle$, which is L-Lipschitz by Lemma I.2, and sum over an orthonormal basis for the second bound. A Poincaré inequality implies exponential concentration of Lipschitz functions (Gromov & Milman, 1983; Bobkov & Ledoux, 1997), (Ledoux, 2001, Sec. 3.1). □

Remark I.5 (Why not Talagrand's convex-distance inequality). Talagrand's inequality (Talagrand, 1995) gives sub-Gaussian tails, but only for *convex* Lipschitz functions of the independent variables. $\hat{o}$ is not convex in e^K because the softmax is not. The Poincaré route needs no convexity, at the cost of exponential instead of Gaussian tails.

Remark I.6 (Bias and the implementation). Theorem I.4 controls fluctuations around $\mathbb{E}\hat{o}$. The bias $\mathbb{E}\hat{o} - o$ comes only from the softmax nonlinearity ($\mathbb{E}e^V = 0$), and Proposition I.3 bounds it by the same T-independent quantity. With Wiener scaling ($\alpha < 1$), $\hat{o}$ is an affine function of the same errors with constants multiplied by $\alpha \le 1$, plus a deterministic shrinkage of the logits. In HybridQuant-G, $\alpha = 1$, keys and values have separate dither tables, and every position below $P = 40960$ has its own dither row (Appendices B and B.1), so the hypothesis of Theorem I.4 holds exactly for every context of up to 40960 tokens, which covers every sequence evaluated. All KV heads of a layer share the dither row of a position; this does not matter here, because the theorem concerns a single head. The initial configuration v1 reused 64 dither rows per (layer, role), so there the hypothesis held exactly only for $T \le 64$. The default HybridQuant has no dither, and Theorem I.4 does not apply to it; Lemma I.2 and Proposition I.3 do, because they only use that every block error lies in its scaled Voronoi cell.

Remark I.7 (Projection biases). With centering (Proposition L.1, whose notation we use: k_t is the key before RoPE and $\mathcal{Q}$ the quantizer), $\hat{v}_t = \mathcal{Q}(\tilde{v}_t) + b_v$ and $\hat{k}_t - \mathrm{RoPE}_t(k_t) = \mathcal{Q}(\tilde{k}_t) - \tilde{k}_t$. The logit errors are therefore those of the centered keys, and because $\sum_t \hat{a}_t = 1$,

$$\hat{o} = \sum_t \hat{a}_t\,\mathcal{Q}(\tilde{v}_t) + b_v.$$

So Lemma I.2, Proposition I.3, and Theorem I.4 apply verbatim to the centered keys and values, with Δ_t^K, Δ_t^V the steps of the centered rows and $V_{\max}$ computed from $\tilde{v}_t$. The shift b_v is deterministic and does not affect the fluctuation bounds. It also cancels in $\hat{o} - o$, since $o = \sum_t a_t\tilde{v}_t + b_v$.

J SUMMARY OF THEORETICAL PROPERTIES

This section collects the properties of HybridQuant and HybridQuant-G in one place and adds the ones not stated elsewhere: inner products, reconstruction error, rate–distortion gaps, correlation, lattice optimality and failure probabilities. Throughout, a row $x \in \mathbb{R}^d$ is rotated to $y = Q_R x$, scaled by its RMS σ, and quantized blockwise with step $\Delta = s\sigma$; $e = \hat{y} - y$ is the rotated-domain error, and $\hat{x} - x = Q_R^\top e$ has the same norm and, for isotropic e, the same law. Table 7 gives an overview.

Table 7: Theoretical properties. P: proved here or in the cited section; M: measured; —: does not hold or not claimed.

Property	HybridQuant (no dither)	HybridQuant-G (dither)	Where
Error independent of the input	—	P	Theorem H.2
Unbiased inner products $\mathbb{E}\langle q, \hat{k}\rangle = \langle q, k\rangle$	— (M: slope 0.99–1.00)	P	Proposition J.1
Inner-product error variance	M	$G_{E_8}\Delta^2\|q\|^2$ (P)	Proposition J.1
Deterministic error bound $\|\hat{x} - x\| \le s\|x\|/\sqrt{8}$	P	P	Proposition J.1
Per-row SNR	design (M)	$1/(G_{E_8}s^2)$ exactly (P)	Proposition C.5
No overload / clipping	P	P	Lemma H.1
Gap to $R(D)$, high resolution	0.146 bit (P, ideal coder)	0.146 bit (P, ideal coder)	Proposition J.3
Gap to $R(D)$, all resolutions	—	$\le$ 0.646 bit (P, ideal coder)	Proposition J.3
Inner-product optimality (one-sided)	via MSE (P)	via MSE (P)	Proposition J.2
Coordinate law close to Gaussian	$R \ge 2$ (P)	$R \ge 2$ (P)	Theorem E.7
Lattice floor on spikes	$R = 3$: $\tilde{O}(1/d)$ (P/S)	same	Theorem G.2
Attention bound independent of T	deterministic (P)	fluctuation + tail (P)	Proposition I.3 and Theorem I.4
Exploits correlation (transform gain)	—	—	Remark J.4

J.1 INNER PRODUCTS, RECONSTRUCTION ERROR AND BIAS

Proposition J.1 (Inner products)**.** *Let $q \in \mathbb{R}^d$ be fixed and k a key row with RMS σ and step $\Delta = s\sigma$ (after centering, Proposition L.1).*

(a) Deterministic bound, all modes. $\|\hat{k} - k\| \le \sqrt{d/8}\,\Delta = s\|k\|/\sqrt{8}$*, hence* $|\langle q, \hat{k}\rangle - \langle q, k\rangle| \le s\|q\|\|k\|/\sqrt{8}$.

(b) Unbiasedness and variance with dither ($\alpha = 1$). *Over the dither,* $\mathbb{E}\langle q, \hat{k}\rangle = \langle q, k\rangle$ *and* $\mathrm{Var}\langle q, \hat{k}\rangle = G_{E_8}\Delta^2\|q\|^2$*, for every k and every number of rounds.*

(c) Wiener scaling biases. *With $\alpha < 1$ (v1),* $\mathbb{E}\langle q, \hat{k}\rangle = \alpha\langle q, k\rangle$.

Proof. (a) Each block error lies in the scaled Voronoi cell $\Delta\mathcal{V}_{E_8}$, whose covering radius is Δ (Appendix C), so $\|e\|^2 \le (d/8)\Delta^2$; the rotation is orthogonal, $\Delta^2 = s^2\|k\|^2/d$, and Cauchy–Schwarz gives the second claim. (b) By Theorem C.4 and Lemma H.1, e is uniform on $\Delta\mathcal{V}^{d/8}$ and independent of k, so $\mathbb{E}e = 0$ and $\mathbb{E}ee^\top = G_{E_8}\Delta^2 I$ (Lemma C.2); $\langle q, \hat{k} - k\rangle = \langle Q_R q, e\rangle$. (c) The reconstruction is $\alpha\hat{y}$, and $\mathbb{E}\hat{y} = y$ by (b). □

Without dither, the error is a deterministic function of the input and can correlate with it, so the estimate is biased in general. The measured least-squares slope of $\langle q, \hat{k}\rangle$ on $\langle q, k\rangle$ is 0.989–1.002 for HybridQuant and 0.999–1.002 for HybridQuant-G, against values as low as 0.886 for TurboQuant-MSE at 2 bits (Table 22). Fixed-rate MSE-optimal scalar codebooks shrink towards zero, which is why TurboQuant adds a QJL residual for unbiased inner products (Zandieh et al., 2026).

Proposition J.2 (One-sided inner products reduce to MSE)**.** *If the query q is independent of the quantization error and isotropic, $\mathbb{E}qq^\top = (\mathbb{E}\|q\|^2/d)I$, then $\mathbb{E}\langle q, \hat{k} - k\rangle^2 = (\mathbb{E}\|q\|^2/d)\,\mathbb{E}\|\hat{k} -$*

$k\|^2$. *Consequently, for a Gaussian source with per-coordinate variance σ^2 coded at R bits per coordinate, no quantizer achieves $\mathbb{E}\langle q, \hat{k} - k\rangle^2 < \mathbb{E}\|q\|^2\sigma^2 2^{-2R}$.*

Proof. Condition on $e = \hat{k} - k$: $\mathbb{E}_q\langle q, e\rangle^2 = e^\top \mathbb{E}[qq^\top]e$. The bound is the Gaussian distortion–rate function $D(R) = \sigma^2 2^{-2R}$. □

In attention the query is not compressed, so the relevant problem is one-sided. Proposition J.2 says that for queries that are not aligned with the error, the best achievable logit error is governed by the MSE of the keys, which is what the rate targeting controls. The two-sided problem, in which both operands are compressed, is analysed by Ordentlich & Polyanskiy (2026) (Appendix K). Without dither, a deterministic quantizer can have errors aligned with a particular query (Ordentlich & Polyanskiy, 2026, Prop. 1); Proposition J.1(b) excludes this for HybridQuant-G.

J.2 Rate–distortion and information-theoretic optimality

Proposition J.3 (Distance from the rate–distortion bound)**.** *Let $G(\Lambda)$ be the normalised second moment of the quantizing lattice.*

(a) High resolution. *An entropy-coded lattice quantizer spends $\frac{1}{2}\log_2(2\pi e G(\Lambda))$ bits per dimension more than the rate–distortion function of a Gaussian source at the same MSE (Gray & Neuhoff, 1998; Zamir, 2014): 0.254 bit for $\mathbb{Z}$, 0.146 bit for E_8 (1.53 and 0.88 dB).*

(b) All resolutions, with dither. *With subtractive dither and a coder conditioned on the dither, the rate exceeds $R(D)$ by at most $\frac{1}{2}\log_2(4\pi e G(\Lambda))$ bits per dimension, for every source and every distortion level (Zamir & Feder, 1992; 1996): 0.754 bit for $\mathbb{Z}$ and 0.646 bit for E_8.*

(c) Fixed-rate scalar quantization*, as in TurboQuant and KIVI, is $\frac{1}{2}\log_2(\sqrt{3}\pi/2) = 0.722$ bit per dimension (4.35 dB) from $R(D)$ for a Gaussian source at high resolution (Panter–Dite; see Gray & Neuhoff (1998)), and 2.74 dB at 2 bits.*

Our coder does not condition on the dither (Appendix B.2), so (b) holds for HybridQuant-G only up to the excess $I(Q;U)$. The side information adds 0.133 bits per element at $d = 128$ (0.80 dB at fixed payload). Figure 4 checks (a) on Gaussian rows: the gap of HybridQuant to $R(D)$, side information included, falls from 2.0 dB at 2.1 bits to 1.70 dB at 5.1 bits, against the limit $0.88 + 0.80 = 1.68$ dB; the integer lattice settles 0.65 dB higher, as $G(\mathbb{Z})/G_{E_8}$ predicts, while TurboQuant-MSE is 3.4–4.7 dB and KIVI ($g = 128$) 5.3–7.5 dB from the bound.

Table 8: Normalised second moments of the best known lattice quantizers in low dimension (Conway & Sloane, 1999, Table 2.3), and the granular gain over scalar rounding. The sphere bound $1/(2\pi e)$ is not attained by any lattice.

	$\mathbb{Z}$	A_2	D_4	E_8	Λ_{24}	bound
G	0.08333	0.08019	0.07660	0.07168	0.06577	0.05855
gain (dB)	0	0.17	0.37	0.65	1.03	1.53
decoding cost	minimal	low	low	low	high	—

Lattice choice. E_8 is the best known lattice quantizer in dimension 8, its blocks tile any d that is a multiple of 8, and its exact decoder costs a few operations per coordinate. Of the remaining 0.88 dB, the Leech lattice Λ_{24} would recover 0.38 dB, but $24 \nmid 128$ and its exact decoders are far more expensive (Appendix O).

Remark J.4 (Correlation and the transform-coding gain). Let the rows have covariance Σ with eigenvalues $\lambda_1, \dots, \lambda_d$. The Hadamard rotation equalises coordinate variances on average, $\mathbb{E}_D[(Q_R \Sigma Q_R^\top)_{ii}] = \operatorname{tr}\Sigma/d$ (Lemma E.4), but, being data-independent, it does not decorrelate. A Karhunen–Loève transform fitted to Σ with optimal bit allocation saves, at high resolution, $\frac{1}{2}\log_2\left(\frac{\text{arithmetic mean of } \lambda_i}{\text{geometric mean of } \lambda_i}\right)$ bits per coordinate. This is the gain that calibrated transform coders such as KVTC (Staniszewski & Łańcucki, 2026) can capture and that HybridQuant forgoes by design. We have not measured it on the models used here.

J.3 Anisotropy measures and failure probabilities

Anisotropy measures. We use four: the peak-to-average ratio $\mu d = d\|y\|_\infty^2/\|y\|_2^2$ (Theorems F.1 and F.2); coordinate and block coherence η, κ (Definition F.4); the Kolmogorov distance of a coordinate to Φ (Theorems E.7 and G.2 and Proposition G.1); and the excess kurtosis and spread of block RMS measured on real rows (Table 20). On Qwen2.5-7B keys, one round lowers the mean μd from 18.8 to 7.9, the excess kurtosis from 3.54 to -0.10 and the Kolmogorov–Smirnov statistic from 0.050 to 0.003; further rounds leave all four unchanged.

Failure probabilities.

1. *Overload:* probability 0. No row is clipped at any design SNR up to 74 dB (Lemma H.1).
2. *Rotation:* for $R \geq 2$, the probability over the $(R-1)$-th sign diagonal that a given row's coordinate law is farther than $0.56\sqrt{2\log(2d/\delta)/d}$ from Φ is at most δ (Theorem E.7). Because the signs are drawn once per layer and role, the statement to use for a whole cache is the expected fraction of rows at distance more than t, which by Markov's inequality is at most $0.894\, d^{-1/2}/t$ ($0.079/t$ at $d = 128$).
3. *Attention deviation (HybridQuant-G):* $\mathbb{P}(|\langle w, \hat{o} - \mathbb{E}\hat{o}\rangle| \geq \epsilon) \leq C\exp(-c\pi\epsilon/(2\bar{\Delta}L))$, uniformly in T (Theorem I.4).
4. *Rate:* escapes occur only for values the Gaussian tables never produced; measured escape rate 0 for every $R \geq 1$ on real rows, and real bitstreams 0.03–0.05% below the accounted size (Appendix B.2).
5. *Decoding:* the code is lossless; every stream decodes bit-exactly (Appendix B.2).

K Relation to the rate–distortion theory of matrix multiplication

Attention computes $QK^\top$ and AV: products in which one operand is compressed and the other is not, or both are. Ordentlich & Polyanskiy (2026) study exactly this problem: compress A and B separately to R bits per entry so that $A^\top B$ can be estimated. Three of their results bear directly on our design.

1. *Deterministic quantizers fail for inner products.* For any deterministic encoders of rate R there exist A, B on the sphere of radius $\sqrt{n}$ with inner-product error $\Omega(n^2)$ instead of $O(n)$ (Ordentlich & Polyanskiy, 2026, Prop. 1). The error vector can align with the other operand. Dither prevents this; it is not only a device for unbiasedness. In our setting the other operand is the query (for keys) and the attention weights (for values). In HybridQuant-G, Theorem C.4 makes the errors independent of both. The default HybridQuant has no dither, and this protection does not apply to it; its only randomness is the shared rotation.
2. *Scalar quantization after rotation loses a* $\log n$ *factor.* After a random rotation, ℓ_∞-normalised scalar quantization (as in QuaRot-style schemes (Ashkboos et al., 2024)) has error larger by about $\frac{2}{3}\ln n$ than a scheme with good lattices, because $\|a\|_\infty$ concentrates around $\sqrt{2\ln n}$ times the RMS (Ordentlich & Polyanskiy, 2026, Eq. (13), Thm. 3). This is the mechanism of Proposition H.3. It is avoided here by RMS scaling, a good lattice and entropy coding, which never clip a row (Lemma H.1); with dither, the distortion is moreover independent of the peak (Theorem H.2).
3. *A benchmark.* For i.i.d. Gaussian matrices the optimal distortion for $A^\top B$ is

$$\Gamma(R) = \begin{cases} 2\cdot 2^{-2R} - 2^{-4R}, & R \geq R^*, \\ 1 - \left(1 - 2\cdot 2^{-2R^*} + 2^{-4R^*}\right)\frac{R}{R^*}, & R < R^*, \end{cases} \qquad R^* \approx 0.906,$$

   per entry, normalised (Ordentlich & Polyanskiy, 2026, Thms. 1–2). Below R^* the optimum time-shares (drops coordinates). Their achievability scheme combines a shared random rotation, dithered nested lattices and MMSE scaling.

HybridQuant-G uses the same family of ingredients (shared rotation and subtractive lattice dither; v1 also used MMSE scaling), while the default HybridQuant keeps the rotation and the lattice and drops the dither (Appendix B.1). What the design adds to their scheme is specific to KV caching:

- causal, per-token operation with no calibration;
- RoPE-commuting key shaping (Appendix D; used in v1);
- a closed-form step for a target SNR (Proposition C.5) and a data-free step search for a target rate (Appendix B.2);
- static entropy coding with exact bit accounting;
- the analysis of structured multi-round Hadamard transforms in Appendices E to G.

The operating points of this paper (2–3 bits) lie above R^*, so time-sharing is not needed. Comparing measured logit errors with $\Gamma(R)$ evaluated on real $\|Q\|_F$, $\|K\|_F$, $\|Q^\top K\|_F$ is a natural further check that we leave to future work.

L Projection-bias centering, and the closest related work

L.1 Projection-bias centering

Some model families add a bias to the key and value projections (Qwen2: $k_t = W_k x_t + b_k$ before RoPE, $v_t = W_v x_t + b_v$). A per-row scale then measures $\|k_t\|$, which the bias can dominate, so the quantization error, proportional to the row norm (Theorem H.2), is spent on a component that is known exactly, since it is a weight. We therefore quantize

$$\tilde{k}_t = \mathrm{RoPE}_t(k_t) - \mathrm{RoPE}_t(b_k), \qquad \tilde{v}_t = v_t - b_v,$$

and add the same terms back after dequantization. $\mathrm{RoPE}_t(b_k)$ is computed with the model's own $\cos$ / $\sin$ tables at position t.

Proposition L.1 (Centering is exact and free)**.** *Let $\mathcal{Q}$ be any quantizer and write $\hat{k}_t = \mathcal{Q}(\tilde{k}_t) + \mathrm{RoPE}_t(b_k)$, $\hat{v}_t = \mathcal{Q}(\tilde{v}_t) + b_v$. Then (i) $\hat{k}_t - \mathrm{RoPE}_t(k_t) = \mathcal{Q}(\tilde{k}_t) - \tilde{k}_t$ and $\hat{v}_t - v_t = \mathcal{Q}(\tilde{v}_t) - \tilde{v}_t$, so every guarantee of Appendices H and I holds with $\tilde{k}_t, \tilde{v}_t$ in place of k_t, v_t; (ii) if $\mathcal{Q}$ is the identity, the model is unchanged; (iii) no side information is needed, because b_k, b_v and the RoPE tables are model parameters.*

Proof. RoPE is linear, so $\mathrm{RoPE}_t(k_t) = \mathrm{RoPE}_t(W_k x_t) + \mathrm{RoPE}_t(b_k)$, and (i) follows by subtraction. (ii) and (iii) are immediate. □

Centering changes what is quantized, not the model. In particular it does *not* discard the logit term $q \cdot \mathrm{RoPE}_t(b_k)$, which varies with t and carries positional information; that term is restored exactly. For values the bias is also harmless to attention itself: $\sum_t a_t(v_t) = \sum_t a_t \tilde{v}_t + b_v$ because $\sum_t a_t = 1$.

Why rotation quantizers without centering fail on Qwen2. The bias vectors are the same for every token and can be large: Liu et al. (2025a) report key-bias entries of absolute value up to 402 in Qwen models (their Appendix C.2.1). The token-specific part of each key, which is what distinguishes one token from another in the attention logits, is much smaller. A quantizer that rotates each vector and scales it by its own norm or RMS, as TurboQuant-MSE and our method do, sizes its quantization step for the whole vector, which is dominated by the bias. At 2–4 bits the resulting error is then larger than the token-specific part it is meant to preserve, the attention logits become noise, and the model's output becomes meaningless. This is a collapse, not a degradation: without centering, TurboQuant-MSE on Qwen2.5-7B has wikitext2 perplexity 11093, 11639 and 4207 at 2.125, 3.125 and 4.125 bits, and Qwen2.5-7B-Instruct retrieves 0 of 88 needles at both 2.125 and 3.125 bits (Table 3), against 66 and 87 of 88 with centering. Centering removes the known bias exactly before quantization and restores it afterwards, so the step is sized for the token-specific part alone. Mistral has no projection biases, so centering changes nothing there. We report every baseline with the same centering, so that the comparisons do not benefit from a broken baseline.

Evidence. Without centering, rotation-based quantizers with per-row scales collapse on Qwen2.5-7B, while Mistral-7B, which has no projection biases, is unaffected: there, TurboQuant with and without centering gives identical perplexities. Table 9 lists the numbers. KIVI is largely insensitive because its per-channel zero point absorbs a constant offset. The released code applies centering by default to our method and reports TurboQuant-MSE without and with the same centering (`turbo-*c`), so that baseline is not disadvantaged.

Prior observations. Related effects have been observed before. Liu et al. (2025a) report huge outliers caused by the biases of the key and value layers of Qwen models, which KV quantization before the bias avoids for KVQuant (Hooper et al., 2024) but which, in their words, is not applicable for the rotation-based QuaRot (Ashkboos et al., 2024). SageAttention (Zhang et al., 2025) observes that keys share a large component common to all tokens and subtracts the mean of K over tokens before 8-bit attention. Our step is narrower and exact. It subtracts the known weights: b_v, and b_k after RoPE, where its contribution depends on the position. The step therefore does not depend on the input, needs no statistics and no side information, and restores the positional logit term $q \cdot \mathrm{RoPE}_t(b_k)$ exactly (Proposition L.1).

Table 9: Qwen2.5-7B, wikitext2 perplexity, 64 windows of 4096 tokens; fp16 is 6.405. Effective bits include all side information.

Method	Bits	No centering	Centering
v1 (initial config.)	2.125	9701.8	7.122
v1 (initial config.)	3.125	8688.5	6.533
TurboQuant-MSE	2.125	11093.2	7.529
TurboQuant-MSE	3.125	11639.0	6.626
KIVI-2 ($g = 32$)	3.0	7.199	

L.2 Closest related work

Three recent works are close enough that the differences should be explicit.

NestQuant (Savkin et al., 2025) quantizes weights, the KV cache and activations of Llama-3-8B to 4 bits with self-similar nested lattices built on the Gosset lattice (E_8), used as a drop-in quantizer for every matrix product. It shares with us the E_8 lattice and the viewpoint of Ordentlich & Polyanskiy (2026). We differ in scope and in coding: we compress only the KV cache, at 2–3 effective bits per element including all side information, and we entropy-code the lattice indices with a static, data-free code model; the rate is set by a data-free search over the step (Appendix B.2) rather than fixed by a codebook size.

HyperQuant (Domb et al., 2026) is the closest. It is an arXiv preprint (2026) that had not been peer-reviewed at the time of writing; we discuss it because it is close. It combines a per-tile randomized Hadamard transform, a low-dimensional lattice (E_8, D_4, A_2 or $\mathbb{Z}$), bit-stripping with Rice coding, and bias correction by subtractive dither for the KV cache, with the operating point set through an SNR. Its ingredients largely coincide with ours, and the two methods share a foundation: the result of Zamir and Feder that subtractive lattice dither makes the error uniform on the Voronoi cell and independent of the input (Theorem C.4). HyperQuant uses it for unbiasedness. Theorem H.2 states the consequence that the distortion does not depend on the input's anisotropy; it holds for any subtractively dithered lattice quantizer that does not overload, so it is not a point of difference. The differences are these:

- our analysis of one, two and three same-axis Hadamard rounds, including the separation between them (Appendices E to G);
- a context-length-independent concentration bound for the quantized attention output (Theorem I.4);
- exact centering of the key and value projection biases (Proposition L.1), which concerns the bias vectors of the projections, not the statistical bias that HyperQuant's dither removes;
- static context-modelled rANS coding instead of Rice coding. Our ablations replace rANS by a Rice coder within one pipeline (`abl-rice`, `a3-rice-2.125`).

We did not rerun their code. The `hyperquant-*` rows in the released configuration apply their coder choice inside our pipeline and are not a reproduction.

NexusQuant (Marques, 2026) is an open-source, calibration-free E_8-lattice KV-cache quantizer. It is not peer-reviewed, and we did not evaluate it.

Among earlier work, TurboQuant (Zandieh et al., 2026) applies a random rotation, an optimal scalar quantizer per coordinate, and an optional 1-bit QJL residual for unbiased inner products, all at

a fixed rate. QuIP# (Tseng et al., 2024) uses randomized Hadamard incoherence and E_8-based codebooks for weights. KIVI (Liu et al., 2024b) quantizes keys per channel and values per token with asymmetric scalar quantizers.

M Comparison with established KV-cache quantizers

Table 10 compares design and theoretical properties, and Table 11 the computational cost per cached vector. Entries for other methods describe the published designs, as summarised in Section 2; "not claimed" means that we are not aware of a statement of that property, not that it fails, and "—" that we did not verify the entry. Our experiments re-implement KIVI and TurboQuant-MSE only (Appendix Q.1).

Table 10: Design and theoretical properties. Rate: whether each vector costs a fixed number of bits (fixed) or a data-dependent number with a fixed average target (variable). Unbiased: $\mathbb{E}\langle q, \hat{k}\rangle = \langle q, k\rangle$ for every fixed query. Input-independent: the error law does not depend on the data.

Method	Calibration or training	Rate	Transform	Quantizer	Coding	Unbiased	Input-indep.
KIVI (Liu et al., 2024b)	none	fixed	none	asymmetric uniform, per channel / per token	fixed-length	not claimed	no
KVQuant (Hooper et al., 2024)	calibration	fixed	none (keys pre-RoPE)	non-uniform scalar + sparse outliers	fixed-length	not claimed	no
QJL (Zandieh et al., 2025)	none	fixed	Gaussian projection	1-bit sign	fixed-length	yes (keys)	no
TurboQuant (Zandieh et al., 2026)	none	fixed	random rotation	optimal scalar (+ QJL residual)	fixed-length	with residual	no
PolarQuant (Han et al., 2026)	none	fixed	random rotation + polar	angles	fixed-length	not claimed	no
QuaRot (KV) (Ashkboos et al., 2024)	none for KV	fixed	Hadamard	uniform, per group	fixed-length	not claimed	no
CQ (Zhang et al., 2024)	calibration	fixed	none	learned multi-channel VQ	fixed-length	not claimed	no
CommVQ (Li et al., 2025)	training	fixed	none	learned VQ commuting with RoPE	fixed-length	not claimed	no
KVLinC (Saxena & Roy, 2025)	training	fixed	Hadamard	scalar + trained correction	fixed-length	not claimed	no
KVTC (Staniszewski & Łańcucki, 2026)	calibration	variable	PCA	DP-allocated scalar	DEFLATE	not claimed	no
CacheGen (Liu et al., 2024a)	profiling	variable	delta to anchor token	layer-wise scalar	arithmetic	not claimed	no
NestQuant (Savkin et al., 2025)	—	fixed	—	nested E_8 lattice	fixed-length	not claimed	not claimed
HybridQuant	none	variable	3 Hadamard rounds	E_8	static rANS	no (bounded)	no
HybridQuant-G	none	variable	3 Hadamard rounds	dithered E_8	static rANS	yes	yes

What the theory gives each family. Fixed-rate scalar quantizers after a rotation (TurboQuant-MSE, QuaRot, KIVI) are, at high resolution, 0.72 bit per coordinate from $R(D)$ for Gaussian coordinates; entropy-coded scalar quantization is 0.254 bit away and entropy-coded E_8 0.146 bit (Proposition J.3). Learned codebooks (CQ, CommVQ) and PCA (KVTC) can exploit correlations and non-Gaussian marginals of a specific model, which a data-free design cannot (Remark J.4); the price is calibration and a rate that depends on how representative the calibration set is. With dither, HybridQuant-G is the only design in the table whose error law is provably independent of the input, which yields unbiased inner products without a residual code and the context-length-independent attention bound of Theorem I.4. Without dither, HybridQuant has a deterministic error bound (Proposition J.1) and, empirically, a least-squares slope within 1% of one (Table 22).

Table 11: Computational cost per cached vector of dimension d (encode; decoding is symmetric), and state that must be stored. b: bits per coordinate; m: projection dimension; g: channels per VQ group.

Component	Arithmetic per vector	Stored state
Dense random rotation	d^2 multiply–adds	$d \times d$ matrix (64 KB fp32 at $d = 128$)
Gaussian projection (QJL)	md multiply–adds	$m \times d$ matrix, or a seed
PCA basis (KVTC)	d^2 multiply–adds	$d \times d$ basis, fitted on calibration data
Learned VQ, groups of g channels	nearest of 2^{bg} centroids per group: $O(2^{bg} d)$	codebooks
Scalar quantization	$O(d)$	per-group scales and zero points
R Hadamard rounds (HybridQuant)	$Rd \log_2 d$ additions, Rd sign flips	Rd sign bits (48 bytes for $R = 3$, $d = 128$)
E_8 nearest point	two D_8 roundings and one comparison per block: $O(d)$	none
Static rANS	$9d/8$ table look-ups and state updates	7 frequency tables of $2A + 2$ entries
Dither (HybridQuant-G)	d additions	$P \times d$ table, or none with a counter-based generator

N Complexity, memory and implementation

N.1 Offline and online work

Table 12 lists every step, whether it runs offline (once, before any data is seen) or online (per cached vector), and its time and memory cost. Per vector means one key or value row of one KV head, $d = 128$ in our models. The offline steps depend only on the lattice, the dither law and the target rate, not on the model: the same tables serve every model and layer.

Table 12: Cost of each step. Operation counts are per vector of dimension d; R rounds, P dither rows, A the per-context alphabet bound (Appendix B.2).

Step	When	Time	Memory
Code tables and step s	offline	30 bisection steps on 2^{16} Monte-Carlo blocks, then 2^{20} blocks; a few seconds on a CPU per (lattice, dither, rate)	7 tables of $2A + 2$ frequencies per role; one float s
Sign diagonals	offline	Rd random bits per (layer, role)	Rd bits, or regenerated from the seed
Dither table (HybridQuant-G)	offline	Pd samples	$P \times d$ floats per (layer, role): 21 MB at $P = 40960$
Centering	online	d subtractions; $\mathrm{RoPE}_t(b_k)$ from the model's cos / sin tables	none (the biases are weights)
R Hadamard rounds	online	$R(d \log_2 d + 2d)$ additions and multiplications (3456 for $R = 3$, $d = 128$)	d floats of working memory
RMS scale	online	$2d$ operations and a square root	16 bits stored per vector
E_8 nearest point	online	per block of 8: two D_8 roundings and a comparison, $O(d)$ in total	none
rANS encoding	online	$9d/8$ table look-ups and state updates	$b \cdot d$ bits per vector, 32 bits per 4096-element stream
rANS decoding	online	$9d/8$ look-ups	state per stream
Dequantization	online	d multiplications ($+d$ subtractions with dither)	none
Inverse rounds and uncentering	online	$R(d \log_2 d + 2d) + d$	d floats of working memory
Attention	online	unchanged	unchanged

Table 23 gives measured GPU times: on an A100, one to three rounds over 4096 tokens $\times$ 8 heads take 0.080, 0.133 and 0.186 ms, and E_8 quantization and dequantization 0.075 and 0.043 ms. The complete quantize–dequantize round trip of our unfused PyTorch pipeline takes 1.42–1.62 ms for $R = 0$–3, so most of its time is spent outside the rotation and the lattice.

N.2 Storing the Hadamard transform, and what the kernels do

The matrix is never stored. $H_{ij} = d^{-1/2}(-1)^{\langle i,j \rangle}$ is a function of the indices, and the fast Walsh–Hadamard transform applies it in $\log_2 d$ butterfly stages of $d/2$ additions and $d/2$ subtractions (Ailon & Chazelle, 2009). The only state of the rotation is the R sign vectors, Rd bits per (layer, role): 48 bytes for $R = 3$ and $d = 128$, against 64 KB for a dense fp32 rotation, and they can be

regenerated from the seed. Our kernels store them as fp32 vectors (1.5 KB), which is convenient but not necessary.

What our kernels do.

1. *All rounds in one kernel.* Each row is loaded once into shared memory, all R rounds run there, and the row is written once. Global memory traffic is therefore independent of R, which is why the third round costs only 0.05 ms (Table 23).
2. *Fused signs and normalisation.* The sign flip is applied as the row enters each round and the $1/\sqrt{d}$ scale as it leaves; no separate pass is made.
3. *One thread per E_8 block.* The Conway–Sloane decoder runs in registers without tables. Codes are stored as doubled int16 coordinates, so half-integer E_8 points are exact integers.
4. *Static tables.* Because the code model does not adapt, code lengths are a gather over the codes and have no sequential dependency; real rANS streams are produced and decoded on the CPU for verification.
5. *Bit-exact CPU and GPU results*, with fused multiply–add contraction disabled, so that every run is reproducible.

What a deployment should add (not implemented).

1. *Rotate the query, not the keys.* Since Q_R is orthogonal, $\langle q, \hat{k}_t \rangle = \langle Q_R q, \hat{y}_t \rangle$, and $\sum_t a_t \hat{v}_t = Q_R^\top \sum_t a_t \hat{y}_t^V$. Keeping the cache in the rotated domain costs one forward transform per query and one inverse transform per output, instead of one inverse transform per cached vector: $O(Rd \log d)$ instead of $O(TRd \log d)$ per query and head. The centering terms are added as $\langle q, \mathrm{RoPE}_t(b_k) \rangle$ per position and b_v once.
2. *Counter-based dither.* A counter-based generator (Salmon et al., 2011) keyed by (seed, layer, role, position) replaces the 21 MB dither table of HybridQuant-G with no memory and gives an independent dither at every position without a period.
3. *Fused decode and attention.* Streams of 4096 elements decode independently, so one warp can decode one stream into shared memory and immediately use it in the attention dot products.

N.3 Choosing the number K of candidate sign sets

The implementation can draw K candidate sign sets per (layer, role) and, for each row, keep the one that minimises the peak $\mu(y)$, at a cost of $\log_2 K$ bits of side information per row and K transforms, which can run in parallel. If a single draw leaves a given row above a peak threshold with probability p, the best of K independent draws does so with probability p^K, so a target failure probability δ requires $K = \lceil \log \delta / \log p \rceil$; the rate cost is $\log_2(K)/d$ bits per element (0.016 for $K = 4$ at $d = 128$). With two or more rounds, Theorem E.7 already makes every row's coordinate law close to Gaussian, so p is small; on real keys and values one round leaves no escapes (Table 20), and best-of-4 changed perplexity by $+0.2\%$ $[-0.0, +0.5]$, not significantly (Table 5). We therefore use $K = 1$. A larger K is worth considering only with a single round on adversarially spiky inputs, or for fixed-rate absmax quantizers, whose error the peak sets directly (Proposition H.3).

O Methods considered and not adopted

Each alternative below was either tested (T) or analysed (A). The common reasons for rejection are that it needs calibration data, that it adds side information or computation without a measurable gain, or that it breaks a guarantee that HybridQuant or HybridQuant-G relies on.

Gaussianisation by Wasserstein-optimal marginal maps (A). The static code assumes Gaussian coordinates. Instead of relying on the rotation alone, one could map each coordinate through a monotone transform that minimises the Wasserstein-2 distance of its law to $\mathcal{N}(0, 1)$. In one dimension this optimal map is $\Phi^{-1} \circ F$, and a moment-based approximation is the Cornish–Fisher expansion (Cornish & Fisher, 1938), which corrects the Gaussian quantile with the skewness and excess kurtosis. We did not adopt it for four reasons. (i) The map needs the cumulants of each row

or channel: estimated per row from 128 coordinates they are noisy and cost side information (two fp16 values add 0.25 bits per element at $d = 128$); estimated per channel they require calibration. (ii) There is little left to correct: after one round the measured excess kurtosis of real keys and values is between -0.10 and -0.04 and the Kolmogorov–Smirnov statistic at most 0.003 (Table 20), and after two rounds Theorem E.7 bounds the distance to Φ for every input. (iii) A nonlinear per-coordinate map destroys the structure the analysis uses: the reconstruction error becomes the lattice error passed through the inverse map's Jacobian, so it is no longer uniform, isotropic or independent of the input (Theorem H.2 and Proposition J.1), and inner products of mapped vectors are not inner products of the originals. (iv) The truncated expansion is not monotone for large skewness or kurtosis and must then be rearranged (Chernozhukov et al., 2010), and its inverse has no closed form.

Learned or data-fitted transforms (A). Learned rotations (Liu et al., 2025b) and PCA bases (Staniszewski & Łańcucki, 2026) can decorrelate and so capture the transform-coding gain of Remark J.4, but they need training or calibration data, and a dense $d \times d$ transform costs d^2 operations per vector and d^2 stored values. A dense random rotation has the same cost without the gain. The Hadamard rounds cost $Rd \log_2 d$ operations and Rd bits (Appendix N).

Better lattices (A). The Leech lattice would recover 0.38 dB more of the 0.88 dB granular gap (Table 8), but 24 does not divide 128, and its exact decoders are far more expensive than the few operations per coordinate of E_8. For head dimensions that are not multiples of 8, D_4 and A_2, the best known lattice quantizers in dimensions 4 and 2 (Conway & Sloane, 1999), are the natural blocks; for dimensions that are not powers of two, a block-diagonal Hadamard transform can be used. Neither extension is evaluated, since all models we test have $d = 128$.

Nested lattices and Voronoi codes (A). They give a fixed rate (Conway & Sloane, 1983; Savkin et al., 2025), but must handle overload (points outside the coarse cell) and lose the shaping gain of entropy coding. An entropy-coded infinite lattice never clips a row (Lemma H.1), and its rate adapts to each vector.

Babai's algorithms (A). Nearest-plane decoding has box-shaped cells and round-off decoding parallelepiped-shaped cells; both have normalised second moment at least $1/12$, so they lose the entire E_8 gain (Proposition C.9 and Remark C.10).

Averaging several rotations (A). The average of several rotations is not a rotation (Proposition E.3) and changes the geometry. Selecting the best of K sign sets per row is the valid alternative; it did not help (T, Appendix N.3).

Other entropy coders (T/A). A Rice coder in place of rANS costs 5.7% perplexity at 2.125 bits in the initial configuration (Table 5). Rice codes are Golomb codes with a power-of-two parameter; general Golomb codes and bit-stripping are simplifications of the same kind that we did not evaluate. Adaptive arithmetic or rANS coding would track the data, but makes the rate data-dependent and every stream's decoding depend on its history, which defeats rate targeting and random access.

Wiener scaling, key shaping and cube dither (T). Each was part of the initial configuration v1 (Table 5). Removing Wiener scaling improved perplexity by 0.8% at 2.125 bits; it shrinks the logits and biases inner products (Proposition J.1(c)). Removing the data-free RoPE-pair key shaping (Appendix D) improved it by 0.5%. Cube dither was no better than Voronoi dither, and it is not a fundamental cell of E_8, so Theorem C.4 does not apply to it (Lemma C.3).

P NUMERICAL EVIDENCE

All numbers in this section are produced by the released code: `scripts/appendix/run_all.sh` for synthetic data and `e8kv.diag_rounds`, `e8kv.run_ppl` and `e8kv.bench` for real models. Every script is seeded, and the kernels are bitwise reproducible across CPU and GPU (Appendix V).

P.1 SYNTHETIC SETUP

Rows have $d = 128$ and there are $T = 4096$ tokens. The bulk is $\mathcal{N}(0, I_d)$. The outlier part Uc_t carries a fraction ρ of the energy: amplitude $\sqrt{\rho d/(1-\rho)}$ on a unit vector c_t. Three outlier patterns are used:

- *RoPE pair*: channels 5 and $69 = 5 + d/2$ with $c_t = (\cos 0.013t, \sin 0.013t)$, a rotating pair;
- *8 colliding*: channels $\{3, 11, \ldots, 59\}$, all $\equiv 3 \pmod 8$, with c_t uniform on S^7;
- *8 random*: eight channels chosen uniformly at random, with c_t uniform on S^7.

The quantizer uses $s = 0.969$, the SNR-knob step for 12 dB with Wiener scaling. The reconstruction is *without* Wiener scaling, so the predicted SNR is $10 \log_{10}(1/(G_{E_8} s^2)) = 11.72$ dB. The static rANS model is built from 2^{18} Gaussian blocks; its design payload rate is 2.183 bits per element. Voronoi dithers are independent across tokens. The logit error is normalised by its Zamir–Feder prediction $G_{E_8}\Delta^2\|q\|^2$ for a fixed Gaussian query, so 1.00 means "exactly as predicted".

Table 13: Dithered E_8 versus the number of rounds R at outlier energy $\rho = 0.98$. κ: block coherence of the outlier subspace (Definition F.4). $\max \mu d$: largest peak-to-average power over tokens. SNR: predicted 11.72 dB. Bits: payload bits per element; the design value is 2.183. Esc.: escape symbols. Logit: normalised logit error. Absmax: a fixed-rate 2-bit per-row absmax scalar quantizer on the same rotated rows.

Outliers	R	κ	$\max \mu d$	SNR (dB)	Bits	Esc. (%)	Logit	Absmax SNR (dB)
RoPE pair	0	16.00	126.1	11.71	1.855	0.77	0.98	−10.22
	1	2.00	4.1	11.72	2.173	0.00	0.99	9.66
	2	2.15	9.3	11.72	2.183	0.00	0.99	6.07
	3	2.12	8.3	11.72	2.184	0.00	1.04	6.89
8 colliding	0	16.00	113.9	11.72	2.015	0.34	0.98	−7.23
	1	8.00	9.3	11.72	2.174	0.00	1.00	7.98
	2	3.67	20.8	11.72	2.184	0.00	0.98	4.64
	3	4.79	16.5	11.72	2.184	0.00	1.00	5.33
8 random	0	16.00	114.5	11.71	2.009	0.30	1.01	−7.23
	1	2.00	9.5	11.72	2.184	0.00	1.02	6.76
	2	4.47	19.8	11.72	2.183	0.00	1.01	5.06
	3	4.04	23.9	11.71	2.183	0.00	1.01	4.86

Reading Table 13. (i) The SNR is 11.71–11.72 dB and the logit ratio 0.98–1.04 for every R, including $R = 0$ (Theorem H.2). (ii) κ after one round equals the collision count of Lemma F.7 exactly (2.00 and 8.00). A second round reduces the adversarial case to 3.67; for non-colliding outliers (8 random) one round is already better than random rotation. (iii) Rate: escapes vanish for $R \geq 1$, and the measured payload equals the design rate only for $R \geq 1$. (iv) The peak $\max \mu d$ is smallest after *one* round, and so is the absmax distortion, as Theorem F.2 and Proposition H.3 predict. This is a property of fixed-rate absmax quantizers, not of ours.

Table 14: Without dither the error depends on the input, and rounds restore the design SNR (11.72 dB). Mean payload bits per element are in parentheses. Dithered SNR is 11.71–11.73 dB in every cell.

Outliers	ρ	$R = 0$	$R = 1$	$R = 2$	$R = 3$
RoPE pair	0.90	11.89 (1.82)	11.81 (2.13)	11.73 (2.14)	11.71 (2.14)
RoPE pair	0.98	16.24 (1.74)	14.39 (2.13)	11.93 (2.14)	11.74 (2.14)
8 colliding	0.90	11.84 (1.97)	11.82 (2.13)	11.72 (2.14)	11.72 (2.14)
8 colliding	0.98	14.66 (1.92)	14.34 (2.10)	11.74 (2.14)	11.71 (2.14)

The one-round values match Theorem E.7(a): 0.3413 for a spike, and at least 0.25 and 0.137 from the atoms for $k = 2$ and $k = 8$. After three rounds the mean peak is 2.82–2.83 σ with standard deviation 0.35 for all four inputs; after two rounds it still depends on the input (2.61–2.84).

Table 15: Shape of one rotated coordinate over $2 \cdot 10^5$ sign draws (exact Sylvester H, $d = 128$). KS: Kolmogorov distance of y_i/σ to $\mathcal{N}(0,1)$; the Monte-Carlo floor is ≈ 0.003. BE: the bound $C_0\mathbb{E}\|z\|_3^3/\|z\|_2^3$ of equation 3. $\mathbb{E}\|y\|_\infty/\sigma$: mean peak (standard deviation in parentheses).

Input	R	KS	BE bound	$\mathbb{E}\|y\|_\infty/\sigma$
Spike ($k = 1$)	1	**0.3424**	0.5600	1.000 (0.000)
	2	0.0359	0.0495	2.840 (0.372)
	3	**0.0105**	0.0788	2.831 (0.350)
Fixed pair ($k = 2$)	1	0.2494	0.3960	1.414 (0.000)
	2	0.0504	0.0700	2.613 (0.388)
	3	0.0147	0.0787	2.824 (0.353)
8 equal spikes	1	0.1372	0.1980	1.857 (0.423)
	2	0.0415	0.0766	2.677 (0.422)
	3	0.0123	0.0786	2.824 (0.353)
Dense Gaussian	1	0.0026	0.0729	2.839 (0.352)
	2	0.0026	0.0786	2.833 (0.349)
	3	0.0024	0.0785	2.834 (0.350)

Table 16: Exact Kolmogorov distance of a spike after two and three rounds. Two rounds: exact binomial law. Three rounds: the law of $Z \mid \eta$ in equation 4 computed exactly by FFT, averaged over the number of η draws shown. The last column is the average of the exact conditional distances, an upper bound on the mixture distance.

d	$R = 2$	$\sqrt{d} \cdot d_{\mathrm{K}}$	$R = 3$	$d \cdot d_{\mathrm{K}}$	η draws	$\mathbb{E}_\eta d_{\mathrm{K}}(Z \mid \eta)$
32	0.06997	0.396	0.03801	1.22	2000	0.03916 ± 0.00031
64	0.04967	0.397	0.01927	1.23	2000	0.01931 ± 0.00014
128	0.03519	0.398	0.00966	1.24	1500	0.00969 ± 0.00008
256	0.02491	0.399	0.00484	1.24	800	0.00485 ± 0.00005
512	0.01762	0.399	0.00229	1.17	300	0.00230 ± 0.00004
1024	0.01246	0.399	0.00125	1.28	120	0.00125 ± 0.00003

Two rounds follow $1/\sqrt{2\pi d}$ exactly, the half-atom of Proposition G.1. Three rounds follow $\approx 1.24/d$ (Theorem G.2); the ratio of the two grows as $\sqrt{d}$ and is $3.6\times$ at $d = 128$.

Table 17: Exact constants of Lemmas G.6 and G.7: minimum far fraction $p_r(\theta)$ and maximum normalised pair covariance over both residues and a 4000-point grid $\theta \in [0.4534/\sqrt{d}, \pi/8]$.

d	$\min_{r,\theta} p_r(\theta)$	$\max_{r,\theta} d\,\lvert\mathrm{Cov}\rvert$
64	0.0769	0.121
128	0.0755	0.117
256	0.0793	0.113
512	0.0767	0.113

Round-off grows roughly like $\sqrt{R}$ and stays more than four orders of magnitude below the quantization noise. It is not a reason to prefer fewer rounds at 2–4 bits per element.

P.2 Real models

`e8kv.diag_rounds` runs the model unquantized and passes the true post-RoPE keys and W_V outputs through every R on the side, so all R see identical inputs (Table 20). Table 21 lists perplexity and effective bits for the rounds sweep and the main methods.

Table 23 in the main text gives the GPU time of one to three FWHT rounds and of the quantize–dequantize round trip of one layer's keys for $R = 0, \ldots, 3$ (`e8kv.bench`). The kernels are unfused reference kernels, and the round trip includes the static-model code length but not rANS encoding, so these numbers support the $O(Rd \log d)$ cost claim, not end-to-end speed-ups.

Table 18: A RoPE outlier pair (channels $5, 69$, random unit coefficients per token). Left: escape rate. Right: block-RMS coefficient of variation, i.e. the spread of the 16 per-block RMS values that one per-row scale must serve; smaller is better.

	Escape rate (%)				Block-RMS CV			
ρ	$R=0$	1	2	3	$R=0$	1	2	3
0.90	0.729	0	0	0	1.476	0.392	0.236	0.220
0.98	0.762	0	0	0	2.143	0.436	0.234	0.214

Table 19: Round-off of the rotation computed *entirely in fp16 arithmetic* (cast after every butterfly), relative to an fp32 reference. The quantizer itself runs in fp32; this is a worst case. Quantization noise at the design point is $G_{E_8} s^2 = 0.0673$ of the signal power.

R	$\|y_{16} - y_{32}\|/\|y_{32}\|$	round-off / signal power	round-off / quantization noise
1	6.5×10^{-4}	4.3×10^{-7}	6.3×10^{-6}
2	9.1×10^{-4}	8.3×10^{-7}	1.2×10^{-5}
3	1.1×10^{-3}	1.2×10^{-6}	1.8×10^{-5}

P.3 Needle retrieval by depth

Figure 3 shows needle retrieval of Table 3 by needle depth and context length, one panel per method. It shows where the misses occur, not only how many: at 2.125 bits the misses of TurboQuant-MSE^c are spread over every context length, while HybridQuant and HybridQuant-G miss only isolated cells, and at 3.125 bits both of our modes find every needle.

Table 20: What the lattice sees on real K/V: rounds sweep (`configs/rounds.yaml`; Qwen2.5-7B, WikiText-2, 8 windows of 4096 tokens).

Method	R	Dither	Role	$\overline{\mu d}$	$\max \mu d$	Block CV	Exc. kurt.	KS	SNR (dB)	Bits	Esc. (%)
r0-2.125	0	voronoi	K	18.78	83.2	0.497	3.54	0.0496	10.39	2.115	0.002
r0-2.125	0	voronoi	V	12.34	123.5	0.291	1.53	0.0149	10.77	2.121	0.008
r1-2.125	1	voronoi	K	7.86	31.7	0.238	-0.10	0.0026	10.39	2.125	0.000
r1-2.125	1	voronoi	V	8.05	30.6	0.240	-0.07	0.0026	10.77	2.125	0.000
r2-2.125	2	voronoi	K	8.11	31.8	0.248	-0.05	0.0022	10.39	2.125	0.000
r2-2.125	2	voronoi	V	8.13	35.5	0.242	-0.05	0.0015	10.77	2.125	0.000
r3-2.125	3	voronoi	K	8.16	32.0	0.242	-0.04	0.0031	10.39	2.125	0.000
r3-2.125	3	voronoi	V	8.15	32.2	0.242	-0.05	0.0012	10.77	2.125	0.000
r0-nd-2.125	0	none	K	18.78	83.2	0.497	3.54	0.0496	10.86	2.108	0.004
r0-nd-2.125	0	none	V	12.34	123.5	0.291	1.53	0.0149	11.11	2.118	0.011
r1-nd-2.125	1	none	K	7.86	31.7	0.238	-0.10	0.0026	10.73	2.124	0.000
r1-nd-2.125	1	none	V	8.05	30.6	0.240	-0.07	0.0026	11.11	2.124	0.000
r2-nd-2.125	2	none	K	8.11	31.8	0.248	-0.05	0.0022	10.73	2.124	0.000
r2-nd-2.125	2	none	V	8.13	35.5	0.242	-0.05	0.0015	11.11	2.124	0.000
r3-nd-2.125	3	none	K	8.16	32.0	0.242	-0.04	0.0031	10.73	2.124	0.000
r3-nd-2.125	3	none	V	8.15	32.2	0.242	-0.05	0.0012	11.11	2.124	0.000
r0-3.125	0	voronoi	K	18.78	83.2	0.497	3.54	0.0496	16.66	3.112	0.025
r0-3.125	0	voronoi	V	12.34	123.5	0.291	1.53	0.0149	17.06	3.121	0.021
r1-3.125	1	voronoi	K	7.86	31.7	0.238	-0.10	0.0026	16.66	3.126	0.000
r1-3.125	1	voronoi	V	8.05	30.6	0.240	-0.07	0.0026	17.06	3.126	0.000
r2-3.125	2	voronoi	K	8.11	31.8	0.248	-0.05	0.0022	16.66	3.126	0.000
r2-3.125	2	voronoi	V	8.13	35.5	0.242	-0.05	0.0015	17.05	3.126	0.000
r3-3.125	3	voronoi	K	8.16	32.0	0.242	-0.04	0.0031	16.66	3.126	0.000
r3-3.125	3	voronoi	V	8.15	32.2	0.242	-0.05	0.0012	17.05	3.126	0.000

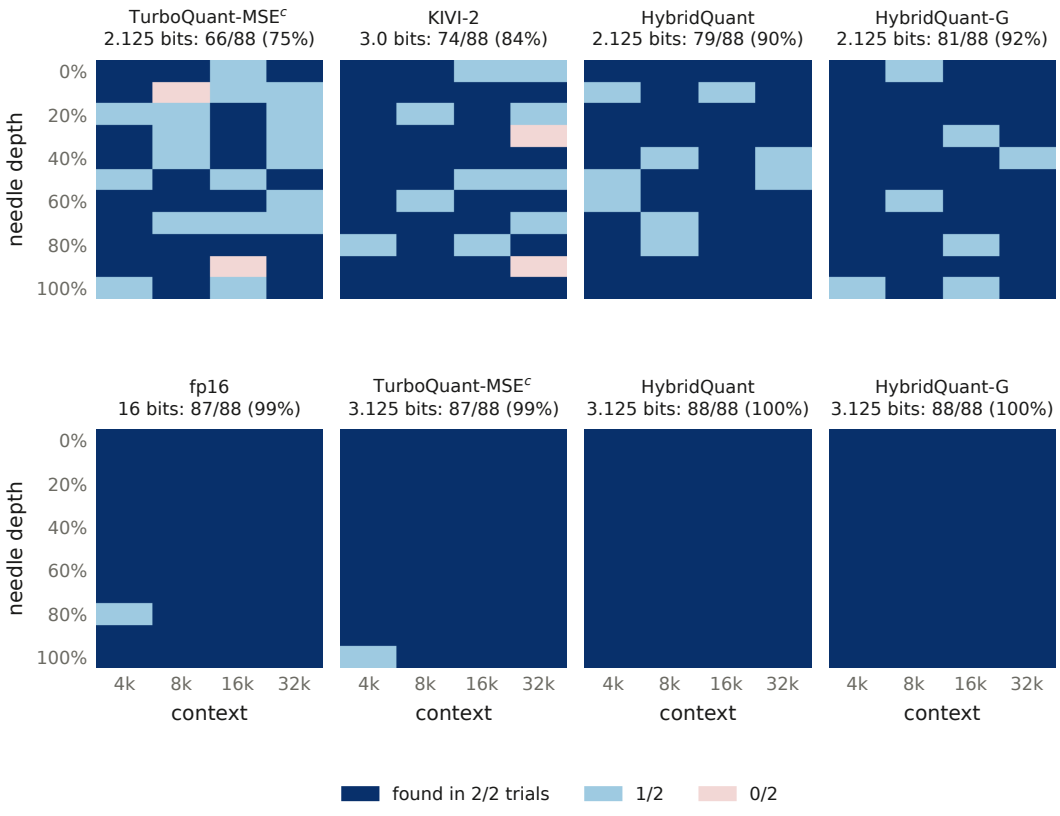


Figure 3: Needle retrieval on Qwen2.5-7B-Instruct by needle depth (rows) and context length (columns). Each cell holds two trials: dark, both needles found; light blue, one; pink, none. Titles give the effective bits and the total over 88 trials. Top: about 2 bits; bottom: fp16 and about 3 bits. TurboQuant-MSE without centering (0/88 at both rates) is omitted.

Table 21: Perplexity and effective bits for the rounds sweep and the main methods (`e8kv.summarize`).

model	method	k_bits	v_bits	avg_bits	ppl_c4	ppl_wikitext2	repro	windows_c4	windows_wikitext2
Qwen/Qwen2.5-7B	r0-nd-2.125	2.107	2.118	2.113	–	7.065	True	–	64/64
Qwen/Qwen2.5-7B	a0-2.125	2.108	2.118	2.113	12.162	6.937	True	64/64	64/64
Qwen/Qwen2.5-7B	r0-2.125	2.115	2.121	2.118	–	7.117	True	–	64/64
Qwen/Qwen2.5-7B	abl-zlattice-2.125	2.123	2.123	2.123	–	7.340	True	–	64/64
Qwen/Qwen2.5-7B	rot-ecsq-2.125	2.123	2.123	2.123	12.366	7.081	True	64/64	64/64
Qwen/Qwen2.5-7B	z3-2.125	2.123	2.123	2.123	12.383	7.077	True	64/64	64/64
Qwen/Qwen2.5-7B	a1-2.125	2.124	2.124	2.124	12.167	6.939	True	64/64	64/64
Qwen/Qwen2.5-7B	r1-nd-2.125	2.124	2.124	2.124	–	7.056	True	–	64/64
Qwen/Qwen2.5-7B	a2-2.125	2.124	2.124	2.124	12.153	6.935	True	64/64	64/64
Qwen/Qwen2.5-7B	r2-nd-2.125	2.124	2.124	2.124	–	7.046	True	–	64/64
Qwen/Qwen2.5-7B	a3-2.125	2.124	2.124	2.124	12.158	6.921	True	64/64	64/64
Qwen/Qwen2.5-7B	a3-nc-2.125	2.124	2.124	2.124	–	11225.914	True	–	64/64
Qwen/Qwen2.5-7B	abl-nodither-2.125	2.124	2.124	2.124	–	7.033	True	–	64/64
Qwen/Qwen2.5-7B	r3-nd-2.125	2.124	2.124	2.124	–	7.033	True	–	64/64
Qwen/Qwen2.5-7B	turbo-2	2.125	2.125	2.125	10522.205	11093.206	True	64/64	64/64
Qwen/Qwen2.5-7B	turbo-2c	2.125	2.125	2.125	13.066	7.529	True	64/64	64/64
Qwen/Qwen2.5-7B	r2-2.125	2.125	2.125	2.125	–	7.128	True	–	64/64
Qwen/Qwen2.5-7B	abl-cubedither-2.125	2.125	2.125	2.125	–	7.135	True	–	64/64
Qwen/Qwen2.5-7B	abl-1round-2.125	2.125	2.125	2.125	–	7.130	True	–	64/64
Qwen/Qwen2.5-7B	r1-2.125	2.125	2.125	2.125	–	7.130	True	–	64/64
Qwen/Qwen2.5-7B	abl-full-2.125	2.125	2.125	2.125	–	7.122	True	–	64/64
Qwen/Qwen2.5-7B	r3-2.125	2.125	2.125	2.125	–	7.122	True	–	64/64
Qwen/Qwen2.5-7B	abl-nommse-2.125	2.125	2.125	2.125	–	7.063	True	–	64/64
Qwen/Qwen2.5-7B	gp-2.125	2.125	2.125	2.125	12.275	6.998	True	64/64	64/64
Qwen/Qwen2.5-7B	abl-noshape-2.125	2.125	2.125	2.125	–	7.089	True	–	64/64
Qwen/Qwen2.5-7B	ours-2.125	2.125	2.125	2.125	12.378	7.122	True	64/64	64/64
Qwen/Qwen2.5-7B	g-2.125	2.125	2.125	2.125	12.251	7.022	True	64/64	64/64
Qwen/Qwen2.5-7B	abl-select4-2.125	2.125	2.125	2.125	–	7.139	True	–	64/64
Qwen/Qwen2.5-7B	abl-nocenter-2.125	2.126	2.125	2.125	–	9701.802	True	–	64/64
Qwen/Qwen2.5-7B	a3-rice-2.125	2.126	2.126	2.126	–	7.198	True	–	64/64
Qwen/Qwen2.5-7B	abl-rice-2.125	2.128	2.127	2.127	–	7.529	True	–	64/64
Qwen/Qwen2.5-7B	kivi-2g128	2.250	2.250	2.250	17.264	10.325	True	64/64	64/64
Qwen/Qwen2.5-7B	a3-k3v2	3.124	2.124	2.624	11.645	6.616	True	64/64	64/64
Qwen/Qwen2.5-7B	turbo-k3v2	3.125	2.125	2.625	11167.246	11471.890	True	64/64	64/64
Qwen/Qwen2.5-7B	turbo-k3v2c	3.125	2.125	2.625	12.024	6.897	True	64/64	64/64
Qwen/Qwen2.5-7B	ours-k3v2	3.126	2.125	2.626	11.736	6.719	True	64/64	64/64
Qwen/Qwen2.5-7B	a3-3.0	2.999	2.999	2.999	11.506	6.540	True	64/64	64/64
Qwen/Qwen2.5-7B	kivi-2	3.000	3.000	3.000	12.484	7.199	True	64/64	64/64
Qwen/Qwen2.5-7B	ours-3.0	3.000	3.000	3.000	11.537	6.550	True	64/64	64/64
Qwen/Qwen2.5-7B	a0-3.125	3.107	3.118	3.112	–	6.513	True	–	64/64
Qwen/Qwen2.5-7B	r0-3.125	3.112	3.121	3.116	–	6.529	True	–	64/64
Qwen/Qwen2.5-7B	a1-3.125	3.124	3.124	3.124	11.474	6.511	True	64/64	64/64
Qwen/Qwen2.5-7B	a3-3.125	3.124	3.124	3.124	11.479	6.509	True	64/64	64/64
Qwen/Qwen2.5-7B	a2-3.125	3.124	3.124	3.124	–	6.512	True	–	64/64
Qwen/Qwen2.5-7B	abl-nodither-3.125	3.124	3.124	3.124	–	6.530	True	–	64/64
Qwen/Qwen2.5-7B	rot-ecsq-3.125	3.124	3.124	3.124	11.514	6.532	True	64/64	64/64
Qwen/Qwen2.5-7B	z3-3.125	3.124	3.124	3.124	11.498	6.526	True	64/64	64/64
Qwen/Qwen2.5-7B	abl-zlattice-3.125	3.124	3.124	3.124	–	6.554	True	–	64/64
Qwen/Qwen2.5-7B	turbo-3	3.125	3.125	3.125	12416.968	11638.951	True	64/64	64/64
Qwen/Qwen2.5-7B	turbo-3c	3.125	3.125	3.125	11.621	6.626	True	64/64	64/64
Qwen/Qwen2.5-7B	r2-3.125	3.126	3.126	3.126	–	6.535	True	–	64/64
Qwen/Qwen2.5-7B	abl-1round-3.125	3.126	3.126	3.126	–	6.528	True	–	64/64
Qwen/Qwen2.5-7B	r1-3.125	3.126	3.126	3.126	–	6.528	True	–	64/64
Qwen/Qwen2.5-7B	gp-3.125	3.126	3.126	3.126	11.481	6.519	True	64/64	64/64
Qwen/Qwen2.5-7B	abl-full-3.125	3.126	3.126	3.126	–	6.533	True	–	64/64
Qwen/Qwen2.5-7B	r3-3.125	3.126	3.126	3.126	–	6.533	True	–	64/64
Qwen/Qwen2.5-7B	abl-nommse-3.125	3.126	3.126	3.126	–	6.515	True	–	64/64
Qwen/Qwen2.5-7B	abl-noshape-3.125	3.126	3.126	3.126	–	6.521	True	–	64/64
Qwen/Qwen2.5-7B	abl-nocenter-3.125	3.126	3.126	3.126	–	8688.481	True	–	64/64
Qwen/Qwen2.5-7B	ours-3.125	3.126	3.126	3.126	11.495	6.533	True	64/64	64/64
Qwen/Qwen2.5-7B	g-3.125	3.126	3.126	3.126	11.470	6.508	True	64/64	64/64
Qwen/Qwen2.5-7B	abl-cubedither-3.125	3.126	3.126	3.126	–	6.529	True	–	64/64
Qwen/Qwen2.5-7B	abl-select4-3.125	3.126	3.126	3.126	–	6.540	True	–	64/64
Qwen/Qwen2.5-7B	abl-rice-3.125	3.126	3.128	3.127	–	6.549	True	–	64/64
Qwen/Qwen2.5-7B	kivi-3g128	3.250	3.250	3.250	11.657	6.629	True	64/64	64/64
Qwen/Qwen2.5-7B	a3-4.125	4.124	4.124	4.124	11.357	6.432	True	64/64	64/64
Qwen/Qwen2.5-7B	turbo-4	4.125	4.125	4.125	5802.139	4206.732	True	64/64	64/64
Qwen/Qwen2.5-7B	turbo-4c	4.125	4.125	4.125	11.401	6.466	True	64/64	64/64
Qwen/Qwen2.5-7B	ours-4.125	4.126	4.126	4.126	11.361	6.434	True	64/64	64/64
Qwen/Qwen2.5-7B	a3-5.0	4.999	4.999	4.999	11.332	6.413	True	64/64	64/64
Qwen/Qwen2.5-7B	kivi-4	5.000	5.000	5.000	11.354	6.424	True	64/64	64/64
Qwen/Qwen2.5-7B	ours-5.0	5.001	5.001	5.001	11.333	6.415	True	64/64	64/64
Qwen/Qwen2.5-7B	fp16	16.000	16.000	16.000	11.322	6.405	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a0-2.125	2.095	2.118	2.107	8.996	5.489	True	64/64	64/64
mistralai/Mistral-7B-v0.3	rot-ecsq-2.125	2.123	2.123	2.123	9.244	5.638	True	64/64	64/64
mistralai/Mistral-7B-v0.3	z3-2.125	2.123	2.123	2.123	9.219	5.597	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a1-2.125	2.124	2.124	2.124	9.027	5.483	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a3-2.125	2.124	2.124	2.124	9.002	5.475	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a2-2.125	2.124	2.124	2.124	8.996	5.466	True	64/64	64/64
mistralai/Mistral-7B-v0.3	turbo-2	2.125	2.125	2.125	10.173	6.204	True	64/64	64/64
mistralai/Mistral-7B-v0.3	turbo-2c	2.125	2.125	2.125	10.173	6.204	True	64/64	64/64
mistralai/Mistral-7B-v0.3	gp-2.125	2.125	2.125	2.125	9.114	5.535	True	64/64	64/64
mistralai/Mistral-7B-v0.3	ours-2.125	2.125	2.125	2.125	9.296	5.703	True	64/64	64/64
mistralai/Mistral-7B-v0.3	g-2.125	2.125	2.125	2.125	9.135	5.532	True	64/64	64/64
mistralai/Mistral-7B-v0.3	kivi-2g128	2.250	2.250	2.250	10.595	6.511	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a3-k3v2	3.124	2.124	2.624	8.433	5.110	True	64/64	64/64
mistralai/Mistral-7B-v0.3	turbo-k3v2	3.125	2.125	2.625	8.659	5.287	True	64/64	64/64
mistralai/Mistral-7B-v0.3	ours-k3v2	3.126	2.125	2.626	8.498	5.170	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a3-3.0	2.999	2.999	2.999	8.381	5.085	True	64/64	64/64
mistralai/Mistral-7B-v0.3	kivi-2	3.000	3.000	3.000	8.867	5.407	True	64/64	64/64
mistralai/Mistral-7B-v0.3	ours-3.0	3.001	3.001	3.001	8.405	5.105	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a1-3.125	3.124	3.124	3.124	8.361	5.067	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a3-3.125	3.124	3.124	3.124	8.369	5.068	True	64/64	64/64
mistralai/Mistral-7B-v0.3	z3-3.125	3.124	3.124	3.124	8.381	5.082	True	64/64	64/64
mistralai/Mistral-7B-v0.3	rot-ecsq-3.125	3.124	3.124	3.124	8.386	5.083	True	64/64	64/64
mistralai/Mistral-7B-v0.3	turbo-3	3.125	3.125	3.125	8.480	5.162	True	64/64	64/64
mistralai/Mistral-7B-v0.3	turbo-3c	3.125	3.125	3.125	8.480	5.162	True	64/64	64/64
mistralai/Mistral-7B-v0.3	gp-3.125	3.126	3.126	3.126	8.370	5.066	True	64/64	64/64
mistralai/Mistral-7B-v0.3	ours-3.125	3.126	3.126	3.126	8.377	5.085	True	64/64	64/64
mistralai/Mistral-7B-v0.3	g-3.125	3.126	3.126	3.126	8.370	5.070	True	64/64	64/64
mistralai/Mistral-7B-v0.3	kivi-3g128	3.250	3.250	3.250	8.440	5.117	True	64/64	64/64
mistralai/Mistral-7B-v0.3	a3-4.125	4.124	4.124	4.124	8.269	5.010	True	64/64	64/64

Q STATISTICAL METHODOLOGY AND BASELINES

The tests below are computed by `e8kv/significance.py` from the per-window and per-trial records of the released runs.

Perplexity. Every method is evaluated on the same 64 windows of 4096 tokens, and the perplexity of a set of windows is the exponential of its mean token negative log-likelihood. To compare methods A and B, we resample the 64 windows with replacement, the same indices for both methods, $10{,}000$ times with a fixed seed, and compute on every resample the relative perplexity difference $\mathrm{PPL}_B/\mathrm{PPL}_A - 1$, where A is the reference (positive values mean that B is worse). We report the 95% percentile interval of this paired bootstrap and a two-sided p-value from the same resamples. The bootstrap measures variation over text windows at a fixed seed; variation over seeds is reported separately.

Needle retrieval. Each method has 88 trials (4 context lengths $\times$ 11 depths $\times$ 2 trials). We report Wilson 95% score intervals for the accuracy. Two methods see the same prompts, so we compare them with an exact McNemar test on the paired trials, i.e. a two-sided binomial test on the trials where exactly one of the two succeeds.

Seeds. The global seed sets the Hadamard sign vectors, the dither tables and the Monte-Carlo code model (Appendix B.2); it also sets the rotation of TurboQuant-MSE, the needle numbers and the LongBench subset. The perplexity windows do not depend on it. HybridQuant and HybridQuant-G at 2.125 and 3.125 bits, and the accuracy-mode variants with $R = 1, 2$ at 2.125 bits, are run with seeds 0–4 on Qwen2.5-7B, WikiText-2. All other runs use seed 0.

Q.1 BASELINE IMPLEMENTATIONS

All baselines are defined in `e8kv/quantizers.py` and run inside the same hooks as HybridQuant: keys after RoPE, values at the output of W_V, and every token quantized once, before any query uses it. Their reported rates include all side information.

KIVI. Keys are quantized per channel over groups of g consecutive tokens, values per token over groups of g consecutive channels. Each group uses an asymmetric uniform quantizer with b-bit codes, whose zero point (the group minimum) and scale ($(\max - \min)/(2^b - 1)$) are stored in fp16, i.e. $32/g$ side bits per element. Keys are grouped within each call of the quantizer, and a trailing partial key group of fewer than g tokens is kept in fp16 and counted at 16 bits per element. With $b = 2$, $g = 32$ gives 3.0 effective bits and $g = 128$ gives 2.25 (`kivi-2g128`), each plus the cost of any fp16 keys. Our implementation differs from the published KIVI (Liu et al., 2024b) in two ways. First, KIVI keeps a full-precision residual of the 128 most recent tokens, while ours keeps in fp16 only the trailing partial group of each call: at most $g - 1$ tokens of a perplexity window or a prompt, plus, during generation, the key of every generated token, because every decoding step is a separate call. Second, in teacher-forced perplexity evaluation a window is a single call, so the statistics of a key's group (its minimum and maximum over g tokens) include up to $g - 1$ later tokens of the same window. This gives KIVI a mild information advantage.

TurboQuant-MSE. Our re-implementation applies one randomized Hadamard round, stores the RMS of each 128-dimensional vector in fp16, and quantizes every normalised coordinate with the Lloyd–Max quantizer for $\mathcal{N}(0, 1)$ at b bits, for $b + 0.125$ effective bits. It includes neither the 1-bit QJL residual of the inner-product variant nor the two-tier split into outlier and regular channels that the TurboQuant paper uses in its KV-cache experiments (Zandieh et al., 2026); every channel gets the same b bits. Hence the label "TurboQuant-MSE (uniform rate)".

Rotated ECSQ. Our pipeline with the integer lattice $\mathbb{Z}$ in place of E_8, $R = 1$, no dither, no Wiener scaling and no key shaping, with the same static rANS coder, rate targeting and centering (`rot-ecsq-*` in `configs/main.yaml`). The matched lattice control `z3-*` in `configs/final.yaml` is the same with $R = 3$, i.e. HybridQuant with $\mathbb{Z}$ in place of E_8.

Centered variants. A superscript c (configurations `turbo-*c`) marks a baseline run with the same projection-bias centering as ours (Appendix L.1). On models without projection biases, such as Mistral-7B, it is identical to the uncentered baseline.

R Inner-product distortion versus bit width

Attention consumes keys only through inner products with queries, so we also measure the inner-product error directly, as a function of the effective bit width. We quantize 8192 synthetic key rows of dimension 128 with each method at $b \in \{2, 3, 4, 5\}$ bits per coordinate, which is $b + 0.125$ effective bits for the rotation-based methods ($b + 0.25$ for KIVI with $g = 128$), with all side information counted, and evaluate 64 Gaussian queries. We report the relative inner-product error $\mathbb{E}[(\langle q, \hat{k}\rangle - \langle q, k\rangle)^2]/\mathbb{E}[\langle q, k\rangle^2]$ and the least-squares slope of $\langle q, \hat{k}\rangle$ on $\langle q, k\rangle$, which is 1 for an estimator without multiplicative bias. Three sources are used: i.i.d. Gaussian rows; Gaussian rows with four fixed channels scaled by 10, as in key outlier channels; and heavy-tailed Student-t_3 rows (`scripts/appendix/ip_distortion.py`; CPU, one seed).

Table 22 and Figure 4 show that our two modes have the smallest inner-product error at every bit width and for every source (they coincide from 4 bits on); relative to the $\mathbb{Z}$ lattice with the same pipeline, E_8 lowers the error by 12–19%, in line with its granular gain (Proposition J.3): 0.654 dB is 14%. Both of our modes have slopes within 1% of one, while TurboQuant-MSE, whose fixed-rate MSE-optimal scalar codebook shrinks towards zero, has slope 0.886 at 2 bits. KIVI with $g = 128$ has slope near one but 2–8 times the error of HybridQuant. On Gaussian rows, the gap of HybridQuant to the rate–distortion bound approaches the value predicted by Proposition J.3 for E_8 with entropy coding plus the side information (1.68 dB), and the integer lattice stays 0.65 dB above it.

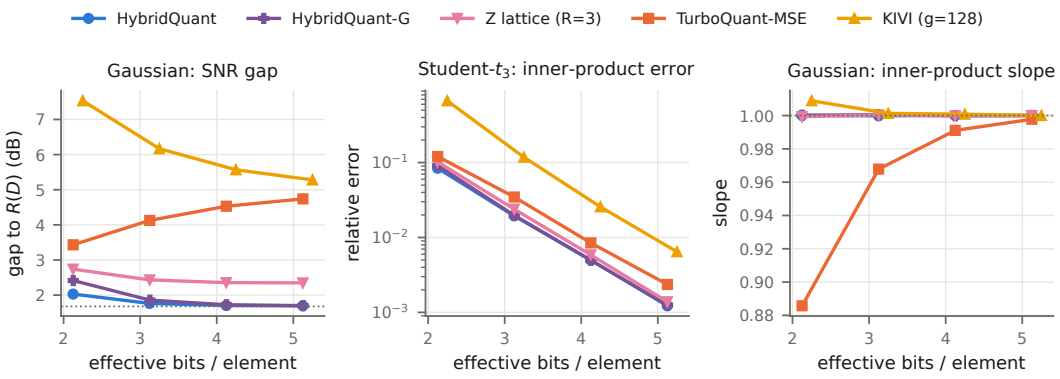


Figure 4: Left: gap between the Gaussian rate–distortion bound and the measured SNR, side information included; dotted: E_8 with ideal entropy coding plus side information (1.68 dB). Middle: relative inner-product error on Student-t_3 rows. Right: least-squares slope of estimated on true inner products (Gaussian rows).

R.1 Comparison in the setting of TurboQuant

Zandieh et al. (2026) evaluate their quantizers on unit-norm vectors against the rate–distortion bounds. Figures 5 and 6 repeat that experiment at our head dimension $d = 128$: 20,000 vectors uniform on the unit sphere, each method at b bits of payload per coordinate (our methods additionally spend 0.133 bits on the fp16 RMS and coder flush; TurboQuant-MSE needs its fp16 norm in practice, which is not charged here), $D_{\mathrm{mse}} = \mathbb{E}\|x - \hat{x}\|^2$ and $D_{\mathrm{prod}} = \mathbb{E}\langle y, x - \hat{x}\rangle^2$ for independent unit queries y, and the Shannon lower bounds 4^{-b} and $4^{-b}/d$ (`scripts/appendix/turbo_style_sim.py`; CPU, about two minutes).

For $b \geq 2$, HybridQuant is closest to the bound at every bit width: at $b = 2$, $D_{\mathrm{mse}} = 0.083$ against 0.090 for HybridQuant-G, 0.098 for the $\mathbb{Z}$ lattice and 0.116 for TurboQuant-MSE (bound 0.0625); at $b = 5$, HybridQuant is 1.23 times the bound (0.9 dB, the E_8 gap of Proposition J.3), while

Table 22: Relative inner-product error (err) and least-squares slope versus bits per coordinate b (8192 rows, 64 queries, $d = 128$).

	$b = 2$		$b = 3$		$b = 4$		$b = 5$	
Method	err	slope	err	slope	err	slope	err	slope
Gaussian rows								
HybridQuant	0.0841	1.000	0.0198	1.000	0.0049	1.000	0.0012	1.000
HybridQuant-G	0.0918	1.000	0.0201	1.000	0.0049	1.000	0.0012	1.000
$\mathbb{Z}$ lattice, R=3	0.0989	0.999	0.0231	1.000	0.0057	1.000	0.0014	1.000
TurboQuant-MSE	0.1161	0.886	0.0341	0.968	0.0093	0.991	0.0024	0.998
KIVI (g=128)	0.2509	1.009	0.0458	1.001	0.0100	1.001	0.0023	1.000
Gaussian rows, 4 outlier channels (10×)								
HybridQuant	0.0832	0.999	0.0196	1.000	0.0048	1.000	0.0012	1.000
HybridQuant-G	0.0908	1.000	0.0200	1.000	0.0049	1.000	0.0012	1.000
$\mathbb{Z}$ lattice, R=3	0.0981	1.000	0.0229	1.000	0.0056	1.000	0.0014	1.000
TurboQuant-MSE	0.0970	0.933	0.0301	0.985	0.0084	0.995	0.0022	0.999
KIVI (g=128)	0.2518	1.011	0.0460	1.001	0.0101	1.001	0.0023	1.000
Student-t_3 rows								
HybridQuant	0.0841	0.989	0.0194	1.002	0.0050	1.000	0.0012	1.000
HybridQuant-G	0.0914	1.002	0.0197	0.999	0.0049	1.000	0.0013	1.000
$\mathbb{Z}$ lattice, R=3	0.1023	1.003	0.0239	1.001	0.0059	0.999	0.0014	1.000
TurboQuant-MSE	0.1205	0.945	0.0347	0.958	0.0085	0.988	0.0024	1.001
KIVI (g=128)	0.6685	1.009	0.1180	1.002	0.0257	1.000	0.0064	1.000

TurboQuant-MSE is 2.5 times the bound, approaching the fixed-rate scalar constant $\sqrt{3}\pi/2 \approx 2.72$. At $b = 1$ the order reverses: TurboQuant-MSE (0.36) beats HybridQuant (0.45) and HybridQuant-G (0.83). At one bit per coordinate the entropy-coded lattice has too few levels to exploit, and the dither adds its full variance; our operating points are 2 bits and above. The histograms at $b = 2$ show the difference that matters for attention: for query–key pairs with fixed $\langle y, x\rangle = \rho$, the inner-product error of TurboQuant-MSE is centred at -0.115ρ (its MSE-optimal codebook shrinks every vector), so its bias grows with the true inner product, whereas the errors of HybridQuant, HybridQuant-G and the $\mathbb{Z}$ lattice stay centred at zero ($|\text{mean}| \leq 2 \times 10^{-4}$) with a spread that does not depend on ρ. TurboQuant removes this bias with a 1-bit QJL residual (TurboQuant-prod); HybridQuant-G is unbiased by construction (Proposition J.1), and HybridQuant is unbiased to within the resolution of this experiment.

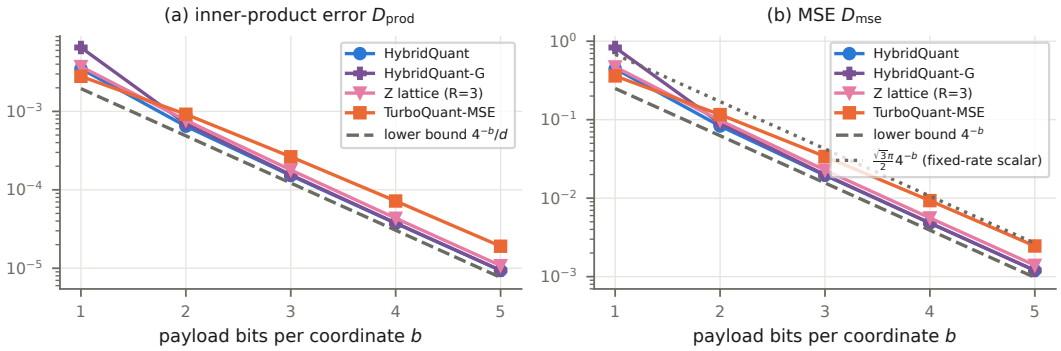


Figure 5: Inner-product error and MSE of unit-norm vectors in dimension 128 against the payload bit width, with the Shannon lower bounds (dashed) and the high-resolution constant of fixed-rate scalar quantization (dotted).

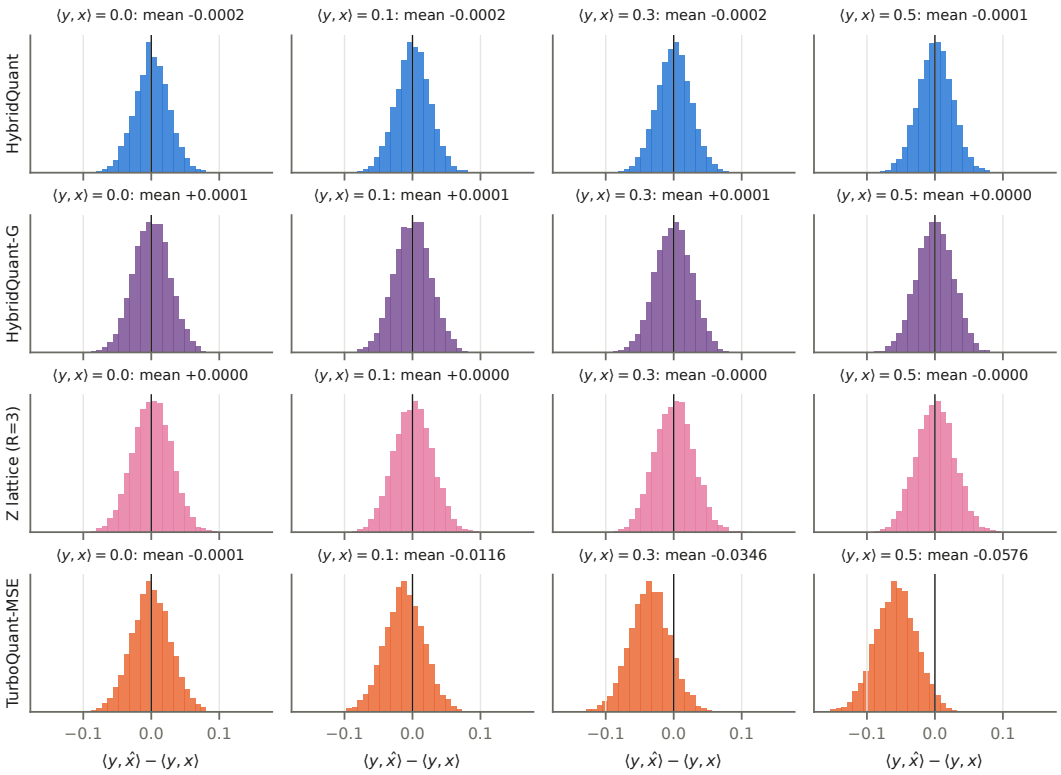


Figure 6: Distribution of the inner-product error $\langle y, \hat{x} \rangle - \langle y, x \rangle$ at $b = 2$ for query–key pairs with fixed true inner product ρ (columns). Only TurboQuant-MSE shifts with ρ.

S ADDITIONAL MEASUREMENTS FOR THE MAIN TEXT

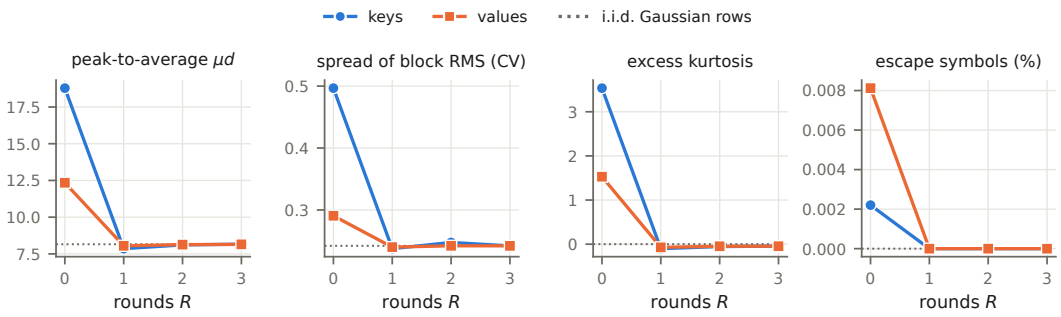


Figure 7: What the lattice sees on real keys and values of Qwen2.5-7B (8 windows of 4096 tokens) against the number of Hadamard rounds R. Dotted: the value of each statistic for i.i.d. Gaussian rows, the law the static code model is built for ($\mu d = 8.16$, CV 0.242, kurtosis 0, no escapes). Without rotation, real rows are spiky (keys: $\mu d = 18.8$, kurtosis 3.5) and produce escape symbols; one round brings every statistic to its Gaussian value, and further rounds leave them there. The third round is therefore not needed on these models; it guards the worst case of Theorem 4.1.

Table 23: GPU time for 4096 tokens $\times$ 8 heads $\times$ $d = 128$ (one layer's keys), unfused reference kernels, NVIDIA A100 80GB PCIe.

Time (ms)	$R = 0$	$R = 1$	$R = 2$	$R = 3$
FWHT rounds only	0	0.080	0.133	0.186
Quantize–dequantize round trip	1.424	1.423	1.529	1.622

T LLAMA-3.1-8B AND A PUBLISHED PROTOCOL

All runs in this section use the code, configurations and harness of the main experiments (`configs/extra.yaml`, `scripts/extra_runs2.sh`), and every number is generated from the run records. Llama weights are the public Llama-3.1-8B and Llama-3.1-8B-Instruct checkpoints. A planned Qwen2.5-14B run and a LongBench-E run of the final default on Qwen did not complete in time and are not reported.

T.1 LLAMA-3.1-8B

Llama-3.1-8B has no projection biases, so centering changes nothing and TurboQuant-MSE is run without it. Its perplexities are in Table 2 and Figures 2 and 8: HybridQuant is significantly better than every other method at every rate, and the margins are larger than on Qwen and Mistral (TurboQuant-MSE has 31.3% [29.8, 33.0] higher wikitext2 perplexity at 2.125 bits, Rot-ECSQ 6.5% [5.9, 7.1]). The measured rate is within 0.0015 bits of its target at every rate.

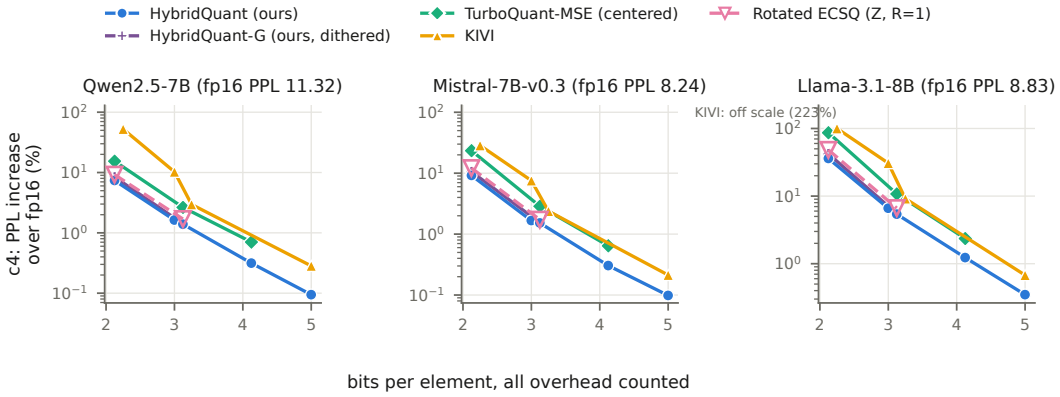


Figure 8: As Figure 2, on C4.

Table 24: Needle-in-a-haystack accuracy on Llama-3.1-8B-Instruct, with the protocol of Table 3 (11 depths $\times$ 2 trials per context length, 88 trials per method, 95% Wilson interval).

Method	Bits	4k	8k	16k	32k	Mean [95% CI]
fp16	16	1.00	1.00	1.00	1.00	1.00 [0.96, 1.00]
KIVI-2 (g=32)	3.0	0.95	1.00	1.00	0.91	0.97 [0.90, 0.99]
TurboQuant-MSE	2.125	0.82	0.55	0.50	0.59	0.61 [0.51, 0.71]
HybridQuant-G	2.125	1.00	0.86	0.91	0.91	0.92 [0.84, 0.96]
HybridQuant	2.125	0.91	0.77	0.95	0.91	0.89 [0.80, 0.94]
TurboQuant-MSE	3.125	1.00	1.00	1.00	1.00	1.00 [0.96, 1.00]
HybridQuant	3.125	1.00	1.00	1.00	1.00	1.00 [0.96, 1.00]

At 2.125 bits HybridQuant retrieves 89% of needles and HybridQuant-G 92%, against 61% for TurboQuant-MSE (exact McNemar $p < 10^{-4}$ for HybridQuant); KIVI-2 at 3 bits retrieves 97% ($p = 0.09$ against HybridQuant, not significant). On LongBench-E HybridQuant averages 49.9 against 40.0 for TurboQuant-MSE at the same rate and 55.3 for fp16; the largest gap is on the synthetic retrieval tasks (50.7 against 21.3).

Table 25: LongBench-E on Llama-3.1-8B-Instruct by category (official metrics, $\times 100$), on a seeded subset of 20 samples per task (260 per method), half the subset of Table 4. Bits are measured.

Method	Bits	SingleQA	MultiQA	Summ.	Few-shot	Synthetic	Code	Average
fp16	16	51.9	52.6	30.7	69.5	60.0	59.7	55.3
TurboQuant-MSE	2.125	35.4	40.9	26.0	64.3	21.3	39.9	40.0
HybridQuant	2.124	48.9	37.8	29.3	67.6	50.7	55.9	49.9

T.2 The protocol of HyperQuant and OCTOPUS

Domb et al. (2026) and Boss et al. (2026) compare KV-cache codecs on Qwen2.5-7B-Instruct-1M at a context of 4096 tokens with the same number of bits for keys and values, and report the perplexity change relative to fp16 on wikitext2 and C4. Domb et al. report that, with every key and value quantized, per-vector codecs diverge on this model even at 4 bits (perplexity about 3500), and both papers keep the keys of the two outer transformer blocks in bf16 (counted in the rate) to prevent it; the TurboQuant and OCTOPUS rows also keep the most recent 32 tokens in full precision. We ran HybridQuant on the same model and context length, on all 73 non-overlapping wikitext2 test windows (72 in Domb et al.) and, as they do, the first 39 C4 windows, with every layer quantized, no residual window, and centering. Table 26 lists our results next to the published ones. The harnesses differ (the two papers report fp16 wikitext2 perplexities of 7.25 and 10.03), so, as Domb et al. do, we compare relative changes, and KV$\times$ is 16 divided by the effective bits, all overhead included.

Table 26: Qwen2.5-7B-Instruct-1M, 4096-token context, perplexity change relative to each harness's own fp16. Reported rows: Domb et al. (2026, Table 12), whose TurboQuant and OCTOPUS rows are from Boss et al. (2026, Table 2); they keep the keys of the two outer blocks in bf16, and "window 32" keeps the last 32 tokens in full precision. Our rows quantize every key and value of every layer, with no window. Collapsed runs are shown by their perplexity.

Source	Method	Bits	KV$\times$	W2 Δ%	C4 Δ%
reported	TurboQuant-MSE (window 32)	4 (nominal)	2.2	+3.1	+1.7
reported	TurboQuant-MSE (window 32)	3 (nominal)	2.6	+8.6	+8.3
reported	TurboQuant-MSE (window 32)	2 (nominal)	3.0	+63.0	+77.4
reported	OCTOPUS (window 32)	4 (nominal)	2.2	+2.7	+1.5
reported	OCTOPUS (window 32)	3 (nominal)	2.5	+7.2	+5.9
reported	OCTOPUS (window 32)	2 (nominal)	2.9	+34.7	+41.5
reported	HyperQuant (no window)	4 (nominal)	3.7	+0.8	+1.0
reported	HyperQuant (no window)	3 (nominal)	4.8	+5.5	+5.7
reported	HyperQuant (no window)	2 (nominal)	6.6	+42.0	+54.3
reported	HyperQuant (window 32)	4 (nominal)	3.6	+0.1	+0.2
reported	HyperQuant (window 32)	3 (nominal)	4.6	+1.8	+1.4
reported	HyperQuant (window 32)	2 (nominal)	6.4	+7.4	+8.1
this work	HybridQuant	4.124	3.9	+0.5	+0.5
this work	HybridQuant	3.124	5.1	+1.8	+1.9
this work	HybridQuant	2.124	7.5	+10.7	+10.4
this work	HybridQuant-G	2.125	7.5	+13.3	+13.0
this work	HybridQuant	1.748	9.2	+30.9	+32.0
this work	TurboQuant-MSE^c (ours)	4.125	3.9	+1.2	+1.1
this work	TurboQuant-MSE^c (ours)	3.125	5.1	+4.5	+4.1
this work	TurboQuant-MSE^c (ours)	2.125	7.5	+26.6	+27.1
this work	HybridQuant, no centering	4.124	3.9	*PPL 4303*	*PPL 5233*
this work	TurboQuant-MSE, no centering	4.125	3.9	*PPL 5758*	*PPL 7156*

Without centering, our quantizer diverges exactly as reported (perplexity 4303 on wikitext2 at 4.125 bits, and 5758 for TurboQuant-MSE), which indicates that the failure these papers work around is the projection bias of Appendix L.1. With centering, HybridQuant needs neither protected layers nor a full-precision window: at 2.125 bits (7.5$\times$) it increases wikitext2 perplexity by 10.7%, against

42.0% for HyperQuant without a window ($6.6\times$), 34.7% for OCTOPUS ($2.9\times$) and 63.0% for TurboQuant ($3.0\times$); HyperQuant with a 32-token full-precision window reaches 7.4% at $6.4\times$. At 3.125 and 4.125 bits HybridQuant is within 1.9% and 0.5% of fp16 at $5.1\times$ and $3.9\times$. Our own TurboQuant-MSE with centering has 26.6% at 2.125 bits, far below the 63.0% reported for TurboQuant without centering but with protected layers and a window.

U WHY THREE ROUNDS: THE CASE, ITS LIMITS AND ITS COST

We use $R = 3$ same-axis rounds. Table 27 collects everything the analysis and the measurements say about the choice. Rounds cost linear time and no memory, so the question is only which properties each additional round buys.

Table 27: What each additional round provides. P = proved, S = lower bound proved and upper bound a proof sketch, M = measured (Appendix P), — = no effect.

Property	$R = 0 \to 1$	$R = 1 \to 2$	$R = 2 \to 3$
Dithered distortion and logit error (Theorem H.2)	— (P)	— (P)	— (P)
No overload (Lemma H.1)	— (P)	— (P)	— (P)
Escapes; measured vs design rate	removed (M)	— (M)	— (M)
Coordinate coherence η	optimal (P, Lemma F.6)	—	—
Input-independent shape (Berry–Esseen)	no: up to 0.341 (P)	yes, $\le 0.894/\sqrt{d}$ in exp. (P)	same bound (P)
Block collisions κ	exact collisions (P)	removed up to log (P)	same bound (P)
Sparse-outlier lattice floor	0.341	$\asymp d^{-1/2}$: 0.0352 at d=128 (P)	$\tilde{\Theta}(d^{-1})$: 0.0097 (S, M)
Block-RMS CV (RoPE pair)	$2.14 \to 0.44$ (M)	$0.44 \to 0.23$ (M)	$0.23 \to 0.21$ (M)
Undithered SNR vs design	partial (M)	reaches design (M)	closer (M)
Peak $\|y\|_\infty$, sparse outliers	reduced (P)	*increased* (P)	— (P)
Cost per row	$+d \log_2 d$ adds	$+d \log_2 d$	$+d \log_2 d$
fp16 round-off / quant. noise	6×10^{-6}	1.2×10^{-5}	1.8×10^{-5} (M)

The case for three.

1. *At least two rounds are necessary.* A calibration-free design uses one code model and one step rule for all rows. That is justified only if the distribution the lattice sees does not depend on the row. After one round it does, by a constant amount (Theorem E.7(a)). One structured round also leaves exact block collisions that no sign choice removes (Lemma F.7); RoPE pairs always collide.
2. *The third round removes the remaining arithmetic structure.* After two rounds a sparse outlier is still supported on a lattice of spacing $2\sigma/\sqrt{d}$, with a provable floor of order $d^{-1/2}$ (Proposition G.1). The third round refines it to order d^{-1} up to a logarithm (Theorem G.2; the upper bound is a proof sketch), a $3.6\times$ reduction at $d = 128$ that grows as $\sqrt{d}$ (Table 16). This is the only separation between two and three rounds that our analysis gives, and it concerns the extreme but realistic case of massive activations (Sun et al., 2024).
3. *It is cheap.* One more fast Walsh–Hadamard transform per row (Table 23), no memory, and no measurable numerical cost (Table 19). The structured random orthogonal features literature likewise uses three HD blocks (Yu et al., 2016; Choromanski et al., 2017).

The limits of the case. Three rounds do *not* lower the dithered distortion (Theorem H.2), do *not* reduce the peak of sparse outliers (Theorem F.2), and are *not* distinguishable from two by generic chaining or by our block-coherence bound (Proposition F.3 and Lemma F.8). Their benefit is in the shape of the distribution, which the static code model and data-free rate targeting rely on. Whether

this changes perplexity on real models is an empirical question that the rounds sweep answers (Tables 20 and 21).

The rule. $R = 2$ is the minimal number of rounds for the universal bound of Theorem E.7(b). We use $R = 3$ because it is the minimal number that also removes the two-round lattice floor for sparse outliers (Proposition G.1 and Theorem G.2; the lower bound is proved, the upper bound is a proof sketch), at the cost of one more FWHT per row (Table 23). If $R = 2$ and $R = 3$ tie on the models tested, the choice rests on this worst-case argument alone, and we do not claim a perplexity gain from the third round.

V REPRODUCIBILITY

The anonymised code is in the supplementary material.

- *Bitwise determinism.* Lattice, dither and rotation kernels use only float32 add/sub/mul with round-half-to-even and FMA contraction disabled (`-ffp-contract=off`, `--fmad=false`, `/fp:precise`). The CPU kernels, CUDA kernels and a pure-PyTorch reference agree bit for bit; this is a unit test. All random tables are drawn from CPU generators seeded by SHA-256, so they are identical on every platform.
- *Crash-safe runs.* Every evaluation unit (window, needle trial) is appended with `fsync` to a run directory named by a hash of the full configuration and of the source of every numerics file. Rerunning a command resumes after the last completed unit, and the final numbers are bit-identical to an uninterrupted run (tested). Each run first evaluates one window twice and aborts if the results differ.
- *Rates are real.* Reported rates are static-model code lengths plus all side bits, as in equation 1. `verify_rans` encodes and decodes the captured codes of one window with the C++ rANS coder, checks that the round trip is exact, and reports the gap between actual and accounted size (Appendix B.2).
- *Commands.* `reproduce.sh` runs build and tests, the kernel benchmark, the main tables, rANS verification, the design ablation, the rounds sweep (`configs/rounds.yaml`), seed variance and needle-in-a-haystack for the configurations of `configs/main.yaml`, `ablation.yaml` and `rounds.yaml`, which include v1. The experiments with the two operating modes (Appendix B.1), the matched controls and the additional baselines are specified in `configs/final.yaml` and run by `scripts/final_runs.sh`: both modes at all rates on Qwen2.5-7B and Mistral-7B, the accuracy-mode rounds sweep and controls, needle retrieval, and seeds 1–4. `e8kv/significance.py` computes the statistics of Appendix Q. `scripts/appendix/run_all.sh` regenerates every synthetic table in Appendix P.

W STATUS OF EVERY CLAIM

Table 28: Status ledger.

Claim	Status	Where
Dither sampler is uniform on the E_8 cell	Proved	Lemma C.3
E_8 cell covariance is $G_{E_8} I$	Proved	Lemma C.2
Error uniform and input-independent (any input)	Proved (cited)	Theorem C.4
SNR knob exact per row	Proved	Proposition C.5
Wiener factor and SNR relation	Proved	Proposition C.6
Skewness 0, kurtosis $O(1/d)$ after rotation	Proved	Proposition C.8
Babai cells (nearest plane and round-off) have $G \geq 1/12$	Proved	Proposition C.9, Remark C.10
Key-shaping exponent $1/4$; exponent $1/2$ gives no gain	Proved (second-moment model)	Proposition D.2

Claim	Status	Where
Per-head logits force block-diagonal transforms	Proved	Proposition E.2
Averaged rotations are not orthogonal	Proved	Proposition E.3
One round: input-dependent shape (0.3413 for a spike)	Proved	Theorem E.7(a)
$R \geq 2$: input-independent Berry–Esseen bound	Proved	Theorem E.7(b)
Peak bounds; $\sqrt{\log d}$ lower bound for $R \geq 2$	Proved (universal L)	Theorem F.1
Three rounds do not reduce sparse peaks	Proved (universal L)	Theorem F.2
γ_2 bounds identical for $R = 1, 3$	Proved	Proposition F.3
Coordinate coherence optimal after one round	Proved	Lemmas F.5 and F.6
Exact Sylvester block collisions	Proved	Lemma F.7
Second round removes collisions up to log	Proved	Lemma F.8
Two-round lattice floor	Proved	Proposition G.1
Three-round rate: lower bound $\Omega(1/d)$, upper bound $O(\log d/d)$	Lower bound proved; upper bound proof sketch	Theorem G.2
Far-fraction and pair-covariance constants	Proof sketch; exact for $d \leq 512$	Lemmas G.6 and G.7, Table 17
Sharp three-round rate is $\Theta(1/d)$	Open (measured $1.24/d$)	Table 16
Three-round result for general inputs	Open	Appendix G
No overload for SNR ≤ 74 dB at $d = 128$	Proved	Lemma H.1
Dithered distortion independent of R and anisotropy (HybridQuant-G); verified on synthetic data	Proved (application of Zamir–Feder)	Theorem H.2, Table 13
Absmax distortion $\propto d\mu$	Proved (uniform-error model)	Proposition H.3
Softmax TV contraction	Proved	Lemma I.1
T-independent Lipschitz constant and worst-case bound (HybridQuant, HybridQuant-G)	Proved	Lemma I.2 and Proposition I.3
T-independent variance and tails of the attention output	Proved; applies to HybridQuant-G	Theorem I.4
Attention bounds with projection-bias centering	Proved	Remark I.7
Escapes vanish and rate matches design for $R \geq 1$	Measured (synthetic)	Tables 13 and 18
Effective rate within 0.003 bits of target for $R \geq 1$; up to 0.012 bits off for $R = 0$	Measured (real models)	Appendix B.2
Code model ignores the dither value, costing $I(Q;U)$ bits per block in HybridQuant-G	Identity; effect on accuracy open (hypothesis)	Appendix B.2
Undithered SNR reaches design only for $R \geq 2$	Measured (synthetic)	Table 14
fp16 round-off negligible for $R \leq 3$	Measured	Table 19
Effect of R on real K/V and perplexity	Measured	Tables 20 and 21
Projection-bias centering is exact and free	Proved	Proposition L.1
Without centering, per-row-scaled rotation quantizers collapse on Qwen2	Measured	Table 9