OCTOPUS: Optimized KV Cache for Transformers via Octahedral Parametrization Under optimal Squared error quantization

Mark Boss Stability AI
Vikram Voleti Stability AI
Simon Donné Stability AI
Shimon Vainer Stability AI

Abstract

The key-value (KV) cache dominates memory bandwidth and footprint in long-context autoregressive inference. Recent rotation-preconditioned codecs (TurboQuant, PolarQuant) show that a structured random rotation followed by a per-coordinate scalar quantizer matched to an analytically tractable marginal is a near-optimal recipe for KV compression. OCTOPUS advances this paradigm through joint quantization of rotated coordinate *triplets*. Each triplet's direction is mapped to a square via an octahedral parameterization, and the two resulting coordinates and the triplet norm are Lloyd-Max quantized against implementation-matched marginals. Optimizing the per-triplet squared error gives a strictly non-uniform bit allocation depending only on the total dimensionality of the keys. We find the finite-dimensional quality optimum with sweeps to be constant on every real decoder we test. The codec is data-oblivious, online, and deterministic given a seed. Across text, video, and audio, OCTOPUS matches or beats every prior rotation codec at every reported bit width and metric, with a lead that grows as bits drop for extreme compression. Furthermore, a fused Triton implementation reconstructs keys on the fly without materializing the uncompressed key, so the codec adds no decode-time bandwidth or latency over the existing dequantization. Project Page: `https://octopus-quant.github.io/`

1 Introduction

Long-context autoregressive inference, such as in large language models (LLM) [9], causal video generation models [39, 48], or audio generation models [31], is dominated by reading the key-value (KV) cache from high-bandwidth memory at every decoding step [12, 25]. KV compression is therefore the primary target for both latency and batch-size optimization, and prior works address it through token eviction [23, 37, 46], per-channel scalar quantization with residuals [17, 20, 27], and more recently rotation-preconditioned quantization codecs [15, 42, 43].

Rotation-based codecs depend on a structured random orthogonal $\boldsymbol{R}$ (typically a sign-flipped Walsh-Hadamard transform due to efficiency [4]) to make the marginal of every rotated key coordinate isotropic and *analytically* known. A 1-D Lloyd-Max quantizer [28, 29] matched to that marginal is then near-optimal at matched bit width. In this way, TurboQuant [43] gets a symmetric Beta marginal, PolarQuant [15] does the analogous construction on recursive polar angles, and the QJL 1-bit residual makes the dot product unbiased at near-zero memory cost [42]. All three quantize one coordinate (or one angle) at a time. OCTOPUS instead quantizes coordinate-*triplets* jointly.

Two observations motivate OCTOPUS. *First*, the rotation pre-conditioning evenly spreads entropy across the coordinates: the norm of a small sub-block carries asymptotically less entropy with rising channel count. We show that a codec that quantizes sub-block norm and direction separately, with non-uniform bit allocation between them, beats the per-coordinate quantizers at matched rate. *Second*,

Preprint.

the octahedral map from computer graphics [5, 10] is an equal-area parameterization of S^2 that can encode a unit 3-vector as two scalars on $[-1, 1]^2$ in $\mathcal{O}(1)$ arithmetic operations, with piecewise-linear encode/decode and a near-uniform Jacobian that makes 1-D Lloyd-Max on the induced marginals a close approximation to true 2-sphere distortion. Therefore, OCTOPUS splits the pre-conditioned signal into triplets, and Lloyd-Max-quantizes the triplet norm and the octahedrally-mapped triplet direction coordinates with non-uniform bit depth. There is no data-dependent calibration or per-vector scale: codebooks depend only on d and the bit budget. Our contributions are:

- **Octahedral triplet direction quantizer** as a KV cache primitive, with implementation-matched norm and direction marginals. The compress-decode pipeline is implemented as fused Triton kernels [6, 32, 34] that reconstruct keys on the fly from packed bit indices and never needs to materialize the full key tensor.
- **An MSE-optimal non-uniform bit split.** A Lagrangian on the per-triplet squared error yields a finite-dimensional stationarity condition that supports the implemented $(b{+}1, b{-}1)$ split at $d{=}128$.
- **Optional 1-bit QJL residual** (OCTOPUS-QJL) that drives the seed-averaged dot-product bias to zero at the cost of one sign bit per rotated coordinate.
- **Generalization beyond LLMs.** Prior rotation-preconditioned KV codecs are evaluated only on language models, but the construction is agnostic to the source of the keys: any autoregressive transformer with attention should benefit. We confirm this empirically. OCTOPUS is the best rotation-based codec at matched bit widths $K{=}V \in \{4, 3, 2\}$ in long-context language modeling (Qwen2.5-7B-Instruct-1M [9]), chunk-wise video diffusion (CausVid [39]), frame-wise causal video forcing [48], and next-scale autoregressive audio [31], with larger gaps at lower bit budgets.

Section 2 situates OCTOPUS in the literature; Section 3 develops the codec; and Section 4 reports end-to-end numbers across the four modalities. Appropriate proofs are found in the Appendix.

2 Related Work

KV-cache compression. Token eviction [3, 23, 26, 37, 46] keeps only tokens that are likely to contribute to future attention. Per-channel scalar quantization with per-token residuals attacks the distribution of individual key coordinates [8, 17, 20, 27, 38, 40, 45]. Sparse coding [22] trades a bigger code table for ultra-low rates. Rotation-preconditioned codecs [2, 15, 16, 33, 35, 42, 43] project keys by a data-oblivious random orthogonal operator so that the marginals fed to the quantizer are analytically known; OCTOPUS belongs to this last family.

Rotation-preconditioned quantization. TurboQuant [43] proves that a random orthogonal rotation makes every coordinate of a unit vector marginally symmetric-Beta on $[-1, 1]$, so the MSE-optimal 1-D Lloyd-Max [28, 29] codebook depends only on (d, b) and lands within a small constant of the Zador-Gersho [13, 41] bound. The structured Walsh-Hadamard transform with random sign flips is the standard fast preconditioner [2, 4, 33]. PolarQuant [15] parameterises the rotated direction recursively in polar coordinates instead. OCTOPUS reuses the Walsh-Hadamard rotation but quantizes blocks of three rotated coordinates jointly via an octahedral direction+norm split, which we show gives strictly lower MSE at matched bit rate.

Unit-direction encodings and unbiased estimators. Octahedral and related equal-area parameterizations of S^2 are the de-facto compact direction encoding in real-time rendering [5, 10]; to our knowledge OCTOPUS is the first use of the octahedral map as a direction quantizer in transformer decoding. Orthogonal to MSE-optimal codecs, QJL [42] shows that a 1-bit Johnson-Lindenstrauss sketch gives an unbiased inner-product estimator at essentially zero memory; we compose it with OCTOPUS under the tag OCTOPUS-QJL. We borrow only the rotation idea from the broader quantization literature on weights [4, 11, 24] and weight+activation quantization [2, 7, 36, 47]; the codec, bit allocation and codebooks are specific to the KV cache and online by construction. Fused attention kernels [6, 32] keep our reconstruction in registers.

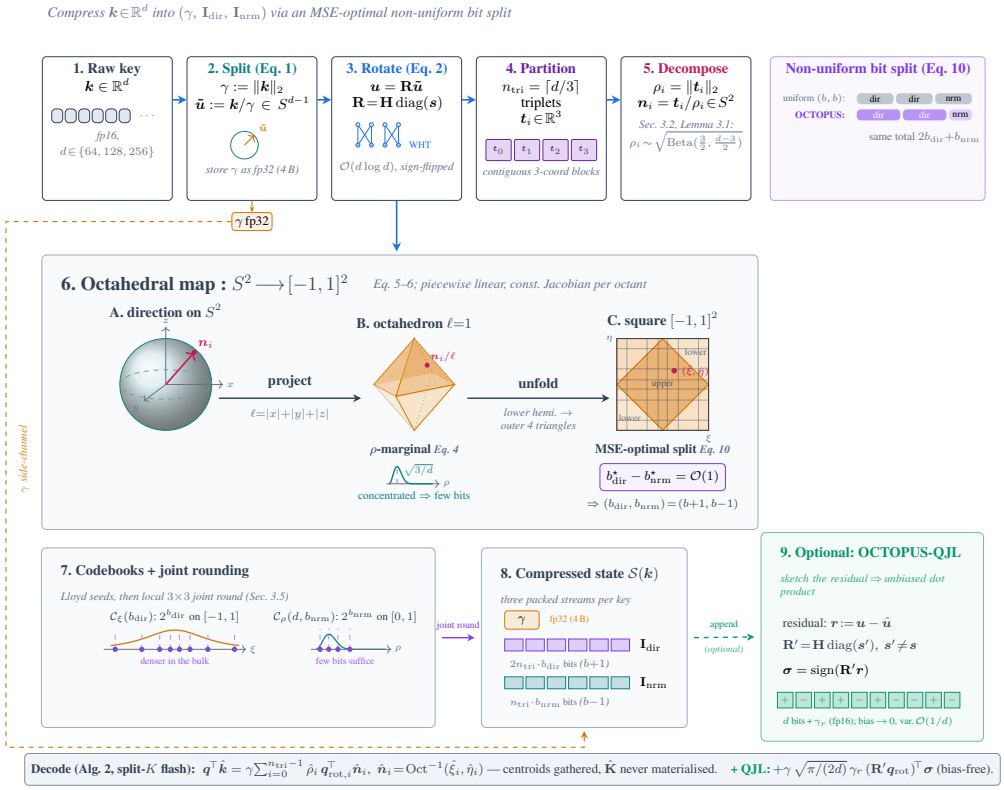


Figure 1: **The OCTOPUS encode pipeline.** Stages 1–5 (top) realise the rotation and triplet decomposition of Sec. 3.1–3.2: a key $\boldsymbol{k}$ is normalised (Eq. 1), preconditioned by a sign-flipped Walsh-Hadamard rotation (Eq. 2), cut into $n_{\text{tri}} = \lceil d/3 \rceil$ triplets, and decomposed into a triplet norm ρ_i and a unit direction $\boldsymbol{n}_i \in S^2$ (Sec. 3.2). Stage 6 (middle) maps each direction onto $[-1, 1]^2$ via the octahedral fold (Eq. 5–6); the analytic triplet-norm marginal (Eq. 4) and empirical oct-coordinate marginal (Eq. 7) drive the Lagrangian bit allocation of Sec. 3.3, whose finite-dimensional stationarity condition (Eq. 10) motivates the implemented $(b{+}1, b{-}1)$ split. Stage 7–8 (bottom) emit the compressed state $\mathcal{S}(\boldsymbol{k}) = (\gamma, \mathcal{I}_{\text{dir}}, \mathcal{I}_{\text{nrm}})$ via the Lloyd-Max codebooks of Sec. 3.4 followed by the local $3{\times}3$ joint rounding of Sec. 3.5. The optional QJL side-car (stage 9, Sec. 3.6) attaches a 1-bit sign sketch of the rotated-frame residual for an ideal-model unbiased dot-product estimator. The bottom strip summarises the fused decode kernel (Alg. 2, App. A).

3 Method

Figure 1 previews the pipeline. Given a key $\boldsymbol{k} \in \mathbb{R}^d$, the OCTOPUS encoder produces a compressed state $\mathcal{S}(\boldsymbol{k}) = (\gamma, \mathcal{I}_{\text{dir}}, \mathcal{I}_{\text{nrm}})$: the global norm, a packed stream of octahedral-coordinate indices, and a packed stream of triplet-norm indices. The decoder reconstructs a lossy $\hat{\boldsymbol{k}}$ inside attention and never materialises $\hat{\boldsymbol{K}}$. We assume d is a power of two, as required by the Walsh-Hadamard transform.

3.1 Rotation preconditioning

We split each nonzero $\boldsymbol{k}$ into magnitude $\gamma \in \mathbb{R}^+$ and direction $\tilde{\boldsymbol{u}} \in S^{d-1}$:

$$\gamma := \|\boldsymbol{k}\|_2, \qquad \tilde{\boldsymbol{u}} := \boldsymbol{k}/\gamma. \tag{1}$$

The magnitude is stored as float32 (4 B per key; $= 0.25$ bpc at $d{=}128$), so almost the entire quantization budget goes to the unit direction. We precondition $\tilde{\boldsymbol{u}}$ by a sign-flipped Walsh-Hadamard transform: with $\boldsymbol{s} \in \{\pm 1\}^d$ drawn once per attention head and $\boldsymbol{H}$ the normalised Hadamard matrix,

$$\boldsymbol{R} := \boldsymbol{H}\,\mathrm{diag}(\boldsymbol{s}), \qquad \boldsymbol{u} := \boldsymbol{R}\,\tilde{\boldsymbol{u}} \in S^{d-1}. \tag{2}$$

$\boldsymbol{R}$ is orthogonal and its inverse runs in $\mathcal{O}(d \log d)$ via an in-place butterfly. Inner products are preserved ($\boldsymbol{q}^\top \boldsymbol{k} = \gamma\,(\boldsymbol{R}\boldsymbol{q})^\top \boldsymbol{u}$), and each coordinate of high-dimensional $\boldsymbol{u}$ has the marginal

$$f(u) \sim (1-u^2)^{(d-3)/2} \qquad u \in [-1,1]. \tag{3}$$

3.2 Triplet decomposition and octahedral coordinates

TurboQuant's MSE baseline quantizes $\boldsymbol{u}$ with a per-coordinate Lloyd-Max [43, Thm. 1]. OCTOPUS instead quantizes *triplets* of rotated coordinates jointly. We partition $\boldsymbol{u}$ into $n_{\text{tri}} = \lceil d/3 \rceil$ contiguous triplets $\boldsymbol{t}_i \in \mathbb{R}^3$, zero-padding the last. For each triplet $\boldsymbol{t}_i$ we again split its norm $\rho_i \in \mathbb{R}^+$ from its direction $\boldsymbol{n}_i \in S^2$. When $\rho_i = 0$, the implementation uses an ϵ-safe divisor and stores a placeholder direction.

Lemma 3.1 (Triplet-norm marginal). *For $\boldsymbol{u}$ uniform on S^{d-1}, $\rho_i^2 \sim \text{Beta}(3/2, (d-3)/2)$, so ρ_i has density*

$$f_\rho(r) = \frac{2r^2\,(1-r^2)^{(d-5)/2}}{B(3/2,(d-3)/2)}, \qquad r \in [0,1]. \tag{4}$$

As $d \to \infty$ the scale $\sqrt{\mathbb{E}[\rho_i^2]} = \sqrt{3/d}$ vanishes, so radial errors contribute less absolute squared error than direction errors.

Octahedral parameterization. We encode $\boldsymbol{n}_i \in S^2$ as two scalars on $[-1,1]^2$ via the octahedral map [5, 10]. With $(x,y,z) = \boldsymbol{n}_i$ and $\ell = |x| + |y| + |z|$, project to the octahedron $\{\ell = 1\}$ via $(p_x, p_y, p_z) = \boldsymbol{n}_i/\ell$, then unfold to a square in $[-1,1]^2$:

$$\text{Oct}(\boldsymbol{n}_i) = (\xi,\eta) \coloneqq \begin{cases} (p_x,\, p_y) & \text{if } p_z \geq 0, \\ \big(\text{sign}(p_x)(1-|p_y|),\, \text{sign}(p_y)(1-|p_x|)\big) & \text{if } p_z < 0. \end{cases} \tag{5}$$

The decoder inverts this: given $(\xi,\eta) \in [-1,1]^2$,

$$\boldsymbol{n}(\xi,\eta) = \frac{(\xi',\eta',\,1-|\xi|-|\eta|)}{\|(\xi',\eta',\,1-|\xi|-|\eta|)\|_2}, \tag{6}$$

with $(\xi',\eta') = (\xi,\eta)$ if $1-|\xi|-|\eta| \geq 0$ and $(\xi',\eta') = (\text{sign}(\xi)(1-|\eta|), \text{sign}(\eta)(1-|\xi|))$ otherwise. The map is a piecewise linear bijection $S^2 \to [-1,1]^2$ with a constant Jacobian per octant [5, 10]. The octahedral fold maps to a *square* code space, so per-coordinate Lloyd-Max on (ξ,η) closely approximates the true 2-sphere distortion, while recursive polar parameterizations [15] need transcendental operators and induce $\sin^{2^{\ell-1}-1}(2\psi)$ angle marginals.

Under the uniform prior on S^2, the octahedral-coordinate marginal is non-uniform. Writing $a = |\xi|$, the marginal induced by the implemented fold is

$$f_\xi(\xi) = \frac{1}{\pi\sqrt{a^2+(1-a)^2}}\left(\frac{1-a}{1-2a+3a^2} + \frac{a}{2-4a+3a^2}\right), \qquad a = |\xi|,\ \xi \in [-1,1], \tag{7}$$

and η shares this marginal by symmetry. Rather than directly evaluate Eq. 7, the implementation trains a Lloyd-Max 1-D codebook on empirical samples of $\text{Oct}(\boldsymbol{n})$, $\boldsymbol{n} \sim \text{Unif}(S^2)$, and shares it between ξ and η.

3.3 MSE-optimal bit allocation

OCTOPUS quantizes each triplet in a total budget of $B_{\text{tri}} = 2b_{\text{dir}} + b_{\text{nrm}}$ bits, where b_{dir} bits go to each octahedral coordinate and b_{nrm} bits go to the triplet norm. We parameterise the allocations around an integer b, with a uniform reference $b_{\text{dir}}{=}b_{\text{nrm}}{=}b$ giving $B_{\text{tri}}{=}3b$. This uniform split is sub-optimal in the *squared-error* sense for any reasonable d.

MSE budget per triplet. Writing the encoder output as $\hat{\boldsymbol{t}}_i = \hat{\rho}_i\,\boldsymbol{n}(\hat{\xi}_i, \hat{\eta}_i)$ and adding/subtracting $\rho_i \hat{\boldsymbol{n}}_i$ gives the bound

$$\|\boldsymbol{t}_i - \hat{\boldsymbol{t}}_i\|_2^2 \leq 2(\rho_i - \hat{\rho}_i)^2 + 2\rho_i^2\,\|\boldsymbol{n}_i - \hat{\boldsymbol{n}}_i\|_2^2, \tag{8}$$

tight up to a $1{+}o(1)$ factor at any reasonable bit width. Under the rotated-sphere prior ρ_i and $\boldsymbol{n}_i$ are independent, so expectations factor. By Panter-Dite high-rate distortion [14, 30], a 1-D

Lloyd-Max quantizer with b bits and source variance σ^2 incurs $D \approx C\sigma^2 4^{-b}$. The first term therefore contributes $2C_\rho\,\sigma_\rho^2\,4^{-b_{\mathrm{nrm}}}$ with σ_ρ^2 to the variance of Eq. 4. The two scalar codebooks on (ξ, η) pull the squared error back to S^2 through the constant-per-octant Jacobian; absorbing that Jacobian and the factor of two into an effective directional variance $\sigma_{\boldsymbol{n}}^2$, the second term contributes $2\mathbb{E}[\rho_i^2]\,C_n\,\sigma_{\boldsymbol{n}}^2\,4^{-b_{\mathrm{dir}}} = (6/d)\,C_n\,\sigma_{\boldsymbol{n}}^2\,4^{-b_{\mathrm{dir}}}$. By Eq. 4, $\sigma_\rho^2 = \mathcal{O}(d^{-1})$ while $\sigma_{\boldsymbol{n}}^2 = \mathcal{O}(1)$, so direction errors remain order-one on S^2 even after the weighting of ρ_i^2.

Lagrangian optimum. Minimizing

$$\mathbb{E}\big[\|\boldsymbol{t}_i - \hat{\boldsymbol{t}}_i\|_2^2\big] \propto 2C_\rho\,\sigma_\rho^2\,4^{-b_{\mathrm{nrm}}} \;+\; (6/d)\,C_n\,\sigma_{\boldsymbol{n}}^2\,4^{-b_{\mathrm{dir}}} \tag{9}$$

subject to $2b_{\mathrm{dir}} + b_{\mathrm{nrm}} = B_{\mathrm{tri}}$ gives

$$b^\star_{\mathrm{dir}} - b^\star_{\mathrm{nrm}} = \log_4\!\left(\frac{3\,C_n\,\sigma_{\boldsymbol{n}}^2}{2d\,C_\rho\,\sigma_\rho^2}\right). \tag{10}$$

Substituting the known $\sigma_\rho^2 = O(d^{-1})$ and $\sigma_{\boldsymbol{n}}^2 = O(1)$, the asymptotic bit gap is independent of key dimensionality d and also notably independent of total bit budget B_{tri}: $b^\star_{\mathrm{dir}} - b^\star_{\mathrm{nrm}} = \mathcal{O}(1)$.

Empirical verification. On synthetic Gaussian keys at d=128 we sweep the diagonal $(b{+}\delta, b{-}\delta)$, $\delta \in \{-2, \ldots, +2\}$, around each uniform reference $b \in \{2, 3, 4\}$. The MSE landscape is sharply convex in δ with minimum at $\delta{=}+1$, i.e. at $(b{+}1, b{-}1)$, for every b tested; relative to uniform (b, b) the implemented split *reduces* MSE by 31–41%, while every other diagonal step *raises* it (by $+44$ to $+73\%$ at $\delta{=}+2$, and by an order of magnitude or more at $\delta{=}-2$). The complete sweep is in App. D; Section 4 shows that the same $(b{+}1, b{-}1)$ split minimizes downstream error across every modality we test.

3.4 Codebooks

Two Lloyd-Max codebooks suffice: $\mathcal{C}_\rho(d, b_{\mathrm{nrm}})$ on $[0, 1]$ matched to Eq. 4, and $\mathcal{C}_\xi(b_{\mathrm{dir}})$ on $[-1, 1]$ matched to the empirical ξ marginal. Both are trained off-line via the standard alternating assignment/update Lloyd-Max iteration to distortion 10^{-10}, are serialized to disk, and are tiny ($\leq 32{+}8$ fp32 centroids per (d, b), ≈ 160 B). They depend only on $(d, b_{\mathrm{dir}}, b_{\mathrm{nrm}})$, without data-dependent calibration.

3.5 Joint rounding of (ξ_i, η_i, ρ_i)

Given the bit split and the codebooks $\mathcal{C}_\xi, \mathcal{C}_\rho$ of Sec. 3.4, the encoder still chooses which code tuple $(\hat{\xi}_i, \hat{\eta}_i, \hat{\rho}_i)$ to emit. Three independent nearest-centroid rounds under Eq. 7 and Eq. 4 are *marginal*-optimal but not *joint*-optimal: the decoder of Eq. 6 is nonlinear in (ξ, η) and multiplicative in ρ, so the product-of-scalar-rounds does not in general minimize

$$\ell(\hat{\xi}_i, \hat{\eta}_i, \hat{\rho}_i) \coloneqq \|\boldsymbol{t}_i - \hat{\rho}_i\,\boldsymbol{n}(\hat{\xi}_i, \hat{\eta}_i)\|_2^2. \tag{11}$$

This is the octahedral analog of the "optimal rounding" pass for tangent-frame codecs in graphics [21], extended to include ρ in the joint.

Simplification. Expanding Eq. 11 with $\|\boldsymbol{n}(\cdot)\|_2 = 1$ gives

$$\ell = \rho_i^2 - 2\hat{\rho}_i\,s_i(\hat{\xi}_i, \hat{\eta}_i) + \hat{\rho}_i^2, \qquad s_i(\hat{\xi}_i, \hat{\eta}_i) \coloneqq \boldsymbol{t}_i^\top \boldsymbol{n}(\hat{\xi}_i, \hat{\eta}_i). \tag{12}$$

For any fixed direction candidate, the optimal $\hat{\rho}_i$ is the $\mathcal{C}_\rho$ centroid nearest to s_i (not to $\|\boldsymbol{t}_i\|_2$), and the joint minimum reduces to maximizing s_i on the direction candidates: $(\hat{\xi}_i, \hat{\eta}_i) = \arg\max s_i$, then $\hat{\rho}_i = \arg\min_{c \in \mathcal{C}_\rho} |c - \mathrm{clip}_{[0,1]}(s_i^*)|$. Direction selection therefore decouples from ρ selection.

Local 3×3 candidate set. The full direction argmax runs over $2^{2b_{\mathrm{dir}}}$ candidates. In practice, the Lloyd scalar seed (i_ξ, i_η) is at most one index away from the joint optimum at every bit width we measured, so OCTOPUS enumerates only the nine candidates $\{(i_\xi + \delta_x,\, i_\eta + \delta_y) \mid \delta_x, \delta_y \in \{-1, 0, 1\}\}$, clamped to the codebook range. Across 10^4 random rotated triplets in d=128 and $b_{\mathrm{dir}} \in \{2, \ldots, 5\}$, this search was *byte-identical* to the full grid search in all buckets at a fraction of the cost (App. E).

Format invariance. Only the *encoder* changes; the bitstream layout, codebooks, and decoder of Eq. 6 are untouched, so joint rounding does not require a decoder change. Every deployed OCTOPUS state (with or without QJL) is decoded by the same fused attention kernel of Sec. 3.6. Algorithm 1 in App. A writes out both variants; we run `local_3x3` as the default throughout Section 4.

3.6 Score path and the optional 1-bit QJL residual

At decode time, the rotated-frame inner product factorizes over triplets:

$$\boldsymbol{q}^\top \hat{\boldsymbol{k}} = \gamma\, \boldsymbol{q}_{\mathrm{rot}}^\top \hat{\boldsymbol{u}} = \gamma \sum_{i=0}^{n_{\mathrm{tri}}-1} \hat{\rho}_i\, \boldsymbol{q}_{\mathrm{rot},i}^\top \hat{\boldsymbol{n}}_i, \tag{13}$$

where $\boldsymbol{q}_{\mathrm{rot}} = \boldsymbol{R}\boldsymbol{q}$ and $\hat{\boldsymbol{n}}_i = \mathrm{Oct}^{-1}(\mathcal{C}_\xi[I_{i,0}^{\mathrm{dir}}], \mathcal{C}_\xi[I_{i,1}^{\mathrm{dir}}])$, $\hat{\rho}_i = \mathcal{C}_\rho[I_i^{\mathrm{nrm}}]$. Only $2n_{\mathrm{tri}}$ direction- and n_{tri} norm-centroid loads are required; $\hat{\boldsymbol{K}}$ never materialized. The encoder and a fused split-K flash decoder are in App. A.

1-bit QJL residual (OCTOPUS-QJL). MSE-optimal scalar quantizers are biased in the dot product [43]. We optionally attach a QJL [42] sketch of the rotated-frame residual $\boldsymbol{r} \coloneqq \boldsymbol{u} - \hat{\boldsymbol{u}}$. With $\boldsymbol{R}' = \boldsymbol{H}\,\mathrm{diag}(\boldsymbol{s}')$ a second rotation with independent seed, we store $\boldsymbol{\sigma} = \mathrm{sign}(\boldsymbol{R}'\boldsymbol{r}) \in \{\pm 1\}^d$ and a residual norm $\gamma_r = \|\boldsymbol{r}\|_2$ (fp16). The QJL estimator $\widehat{\boldsymbol{q}_{\mathrm{rot}}^\top \boldsymbol{r}} = \sqrt{\pi/(2d)}\,\gamma_r\,(\boldsymbol{R}'\boldsymbol{q}_{\mathrm{rot}})^\top \boldsymbol{\sigma}$ is unbiased under the ideal QJL model, with variance $\mathcal{O}(1/d)$; the implementation uses the same scaling with a structured WHT rotation and an fp16-rounded γ_r. The corrected score is $\boldsymbol{q}^\top \hat{\boldsymbol{k}} + \gamma\, \widehat{\boldsymbol{q}_{\mathrm{rot}}^\top \boldsymbol{r}}$.

4 Experiments

We compare OCTOPUS and OCTOPUS-QJL against three rotation-preconditioned codecs sharing the same Walsh-Hadamard rotation, V codec, and residual window: **TurboQuant-MSE** [43] (per-coordinate Lloyd-Max), **TurboQuant-QJL** [42, 43] (MSE stage + 1-bit JL residual), and **PolarQuant** [15] (recursive polar). The only variable across rows is the K codec, and every comparison is matched at the same symmetric $K{=}V$ bit width.

Modalities. (i) A *synthetic probe* of isotropic Gaussian keys at $d{=}128$, the regime in which the rotation-Beta-Lloyd baseline is provably near-optimal. (ii) Long-context language modelling with `Qwen2.5-7B-Instruct-1M` [9]: 7B parameters, GQA [1], 28 layers, $d_h{=}128$, 1M native context. (iii) Two Wan-1.3B autoregressive video DiTs at $d_h{=}64$ and 30 blocks: chunk-wise CausVid [39] (3-frame chunks) and frame-wise Causal Forcing [48]. (iv) The 16-block next-scale autoregressive audio model AAR [31]. Default recipe: short residual window of native-precision tokens/frames/scales and value-side group size $\in \{16, 32\}$. The video and audio cross-codec rows use the unprotected default; the LLM cross-codec rows use the boundary-1 K recipe described in Sec. 4.2 as a setup prerequisite. Compression ratios are (fp16 KV bytes)/(compressed KV bytes).

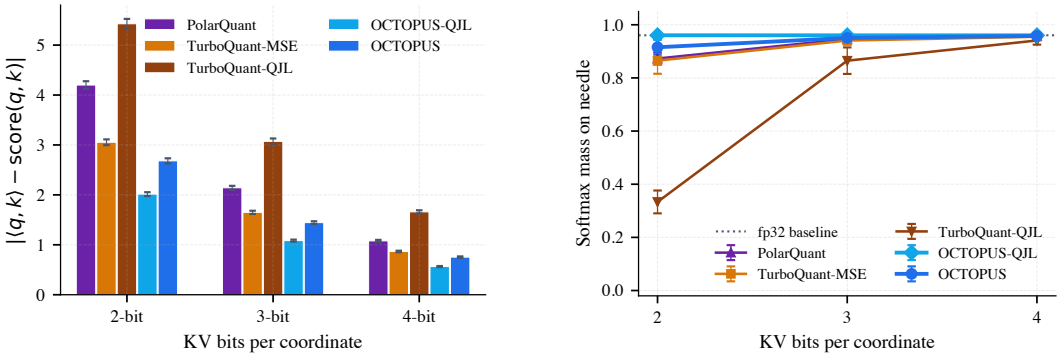
4.1 Synthetic rate-quality and needle retrieval

We draw $n{=}1024$ Gaussian keys and 16 Gaussian queries at $d{=}128$ and average over 64 seeds, reporting reconstruction cosine, per-coord MSE, and inner-product (IP) absolute error $|\boldsymbol{q}^\top \boldsymbol{k} - \mathrm{score}(\boldsymbol{q}, \boldsymbol{k})|$ with each codec's paper-claimed estimator (cf. TurboQuant Fig. 1–2 [43]). For needle-in-a-haystack we plant one key in $T{=}2048$ Gaussian distractors with 10% noisy query and report softmax mass on the needle, averaged over 128 seeds (the fp32 baseline concentrates 0.960).

Table 1 and Fig. 2: OCTOPUS has the best reconstruction fidelity of any rotation codec at every bit width, with MSE $1.3\times$ below the per-coordinate-optimal TurboQuant-MSE at $b{=}4$ and $2.4\times$ below PolarQuant at $b{=}2$. OCTOPUS-QJL drives the IP error $3\times$ below TurboQuant-QJL at a matched rate (the latter spends one bit on its stage-1 quantizer, leaving the reconstruction one bit worse). On the synthetic needle, OCTOPUS-QJL tracks the fp32 baseline to within 0.001; at $b{=}2$, OCTOPUS preserves 0.92 of the softmax mass vs. 0.86/0.87/0.33.

4.2 Long-context language modelling (Qwen2.5-7B-Instruct-1M)

Following Zandieh et al. [43, §5] we report WikiText-2 and C4 perplexity (PPL) (512-token blocks, 8 chunks) and a multi-key needle-in-a-haystack sweep [18, 19] (4k–128k context; 4+1 needles with random 8-char magic values, exact-match scoring). Recipe: residual window 32, V group size 32, K held at fp16 on both boundary blocks ("boundary-1")—a stability prerequisite, not a contribution: every rotated codec diverges to PPL $>10^3$ without it. All Table 2 rows share this setup.



(a) IP absolute error (d=128)

(b) Synthetic needle retrieval

Figure 2: **Synthetic fidelity.** (a) OCTOPUS-QJL is best at every bit width; OCTOPUS alone beats every non-QJL baseline. (b) OCTOPUS-QJL tracks fp32 to within 0.001; TurboQuant-QJL drops to near-uniform at b=2.

Table 1: **Synthetic reconstruction fidelity at $d = 128$.** Isotropic Gaussian keys/queries, averaged over 64 seeds. Reconstruction MSE per coord. Best per bit-width block is **bold**, runner-up underlined.

bits	codec	cos (↑)	MSE (↓)	IP abs err (↓)
2	TurboQuant-MSE	0.9406	0.1161	3.054
2	TurboQuant-QJL	0.7994	0.3610	5.427
2	PolarQuant	0.8902	0.2197	4.200
2	**OCTOPUS**	**0.9547**	**0.0897**	2.682
2	**OCTOPUS-QJL**	**0.9547**	**0.0897**	**2.015**
3	TurboQuant-MSE	0.9831	0.0340	1.650
3	TurboQuant-QJL	0.9406	0.1161	3.072
3	PolarQuant	0.9715	0.0571	2.142
3	**OCTOPUS**	**0.9871**	**0.0260**	1.444
3	**OCTOPUS-QJL**	**0.9871**	**0.0260**	**1.084**
4	TurboQuant-MSE	0.9954	0.0094	0.866
4	TurboQuant-QJL	0.9831	0.0340	1.660
4	PolarQuant	0.9928	0.0145	1.079
4	**OCTOPUS**	**0.9965**	**0.0071**	0.753
4	**OCTOPUS-QJL**	**0.9965**	**0.0071**	**0.565**

Quality. OCTOPUS leads every rotation codec at every bit width (Table 2). In b=4 the WikiText-2 gap is modest (+2.7% vs. +3.1/4.4/8.0%); in b=2 the separation is decisive (+34.7% vs. +63/187/772%).

Needle-in-a-haystack. Multi-key random-value retrieval (4k–128k context, 4 samples per cell; App. H). At b=4 all codecs reach 1.00. At b=3, OCTOPUS holds 1.00; PolarQuant drops to 0.86 average. At b=2 (Fig. 3), only OCTOPUS/0.81 and OCTOPUS-QJL/0.83 retain recall; PolarQuant and TurboQuant-QJL collapse (0.04/0.01), tracking their perplexity divergence.

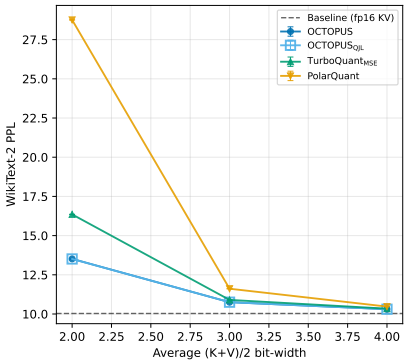
4.3 Autoregressive video and audio

Setup. The video experiments use two Wan-1.3B autoregressive DiTs with 30 blocks, d_h=64, and bf16 activations: CausVid [39], which generates in 3-frame chunks, and Causal Forcing [48], which advances frame by frame. We compress the attention KV cache during generation with a residual window of one native-precision frame, value group size g=32, and no boundary-block protection. For each model and bit width, every codec is run on the same 100 prompts with byte-identical initial noise; the reported deltas therefore isolate the codec rather than prompt or sampling variation. We measure LPIPS [44], PSNR, SSIM, and latent cosine against the uncompressed rollout.

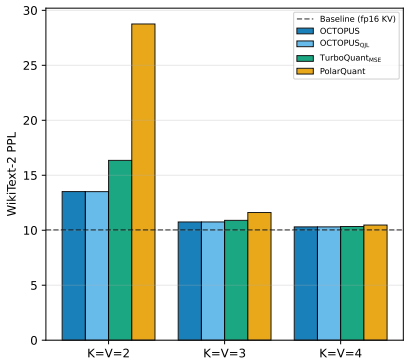
The audio experiment uses AAR [31], a 16-block next-scale autoregressive model. We follow the released CLAP-conditioned inference path: 100 random 10 s AudioSet-20k clips are encoded by CLAP and used as conditioning, while the model generates the corresponding audio continuation/sample

Table 2: **Long-context LM on `Qwen2.5-7B-Instruct-1M`.** WikiText-2 / C4 perplexity at context 4096, symmetric $K{=}V$. Every row uses the same recipe (residual window 32, V group size 32, K-side protection on the outer transformer block at each end—a stability prerequisite for this model, not a contribution). Deltas are vs. fp16. KV× = fp16 bytes / compressed bytes. Best per bit-width block is **bold**, runner-up underlined.

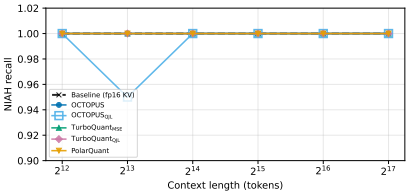
bits	codec	WikiText-2	Δ%	C4	Δ%	KV×
–	fp16 baseline	10.033	0.00	12.701	0.00	1.0×
4	TurboQuant-MSE	10.340	+3.1	12.921	+1.7	2.2×
4	TurboQuant-QJL	10.836	+8.0	13.699	+7.9	2.2×
4	PolarQuant	10.473	+4.4	13.091	+3.1	2.2×
4	**OCTOPUS**	**10.306**	**+2.7**	12.896	**+1.5**	2.2×
4	**OCTOPUS-QJL**	**10.306**	**+2.7**	**12.893**	**+1.5**	2.0×
3	TurboQuant-MSE	10.899	+8.6	13.761	+8.3	2.6×
3	TurboQuant-QJL	15.093	+50.4	20.308	+59.9	2.5×
3	PolarQuant	11.612	+15.7	14.716	+15.9	2.6×
3	**OCTOPUS**	**10.753**	**+7.2**	**13.446**	**+5.9**	2.5×
3	**OCTOPUS-QJL**	10.754	**+7.2**	13.474	+6.1	2.3×
2	TurboQuant-MSE	16.354	+63.0	22.536	+77.4	3.0×
2	TurboQuant-QJL	87.490	+772.0	184.034	+1349.0	3.0×
2	PolarQuant	28.759	+186.6	61.486	+384.1	3.0×
2	**OCTOPUS**	13.517	**+34.7**	17.976	+41.5	2.9×
2	**OCTOPUS-QJL**	**13.511**	**+34.7**	**17.955**	**+41.4**	2.6×



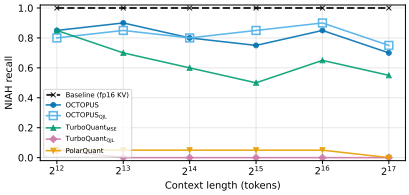
(a) PPL (↓) vs. bits



(b) ΔPPL (↓) per bit width



(c) NIAH recall (↑) ($b{=}4$)



(d) NIAH recall (↑) ($b{=}2$)

Figure 3: **Qwen2.5-7B rate-quality and needle recall.** OCTOPUS does not collapse at $b{=}2$ on either PPL or retrieval; at $b{=}4$ all codecs retain baseline recall.

under compressed KV. The cache recipe matches the video sweep except for the autoregressive unit and group size: residual window one native-precision scale, V group $g{=}16$, and no per-layer protection. We report LSD, log-mel MSE, SNR, and latent cosine against the uncompressed AAR output.

Findings. Table 3 reports per-prompt min/μ/max. At $b{=}4$ all codecs overlap ($\leq 3\%$). At $b{=}2$ the picture changes sharply: on Causal Forcing, TurboQuant-QJL reaches a worst-case LPIPS of 1.00 and a mean of 0.82—effectively random noise—while OCTOPUS stays at 0.58/0.82 (min/max). On audio, the 10 s AudioSet-conditioned sweep is forgiving at $b{=}4$ (all codecs lie within 0.19 dB LSD), but separates sharply at $b{=}2$: TurboQuant-MSE, TurboQuant-QJL, and PolarQuant rise to

Table 3: **Compressed-KV video and audio, symmetric $K{=}V$.** Per-prompt min / μ / max across all prompts. CausVid: residual window 1 frame; Causal Forcing: 1 frame, frame-wise; AAR: 100 random 10 s AudioSet-20k clips as CLAP-audio conditioning, residual window 1 scale, $g{=}16$. KV$\times$ is the video ($g{=}32$) compression ratio. Mean (μ) is **bold** for best, underlined for runner-up per bit-width block.

			CausVid						Causal Forcing						AAR (audio)					
			LPIPS↓			PSNR↑			LPIPS↓			PSNR↑			LSD↓			SNR↑		
b	codec	KV×	min	μ	max	min	μ	max	min	μ	max	min	μ	max	min	μ	max	min	μ	max
4	TurboQuant-MSE	2.4×	0.008	0.045	0.130	17.4	26.5	39.5	0.140	0.334	0.637	9.2	14.6	21.8	0.0	6.4	9.9	-2.1	2.1	120.0
4	TurboQuant-QJL	2.4×	0.014	0.096	0.265	15.5	22.6	34.8	0.213	0.421	0.663	8.1	13.1	18.6	0.0	**6.2**	10.0	-2.4	1.8	120.0
4	PolarQuant	2.4×	0.006	**0.037**	0.124	19.5	**27.8**	42.2	0.113	**0.301**	0.582	11.1	**15.4**	24.2	0.0	6.3	9.6	-2.2	**2.2**	120.0
4	**OCTOPUS**	2.3×	0.005	0.038	0.115	19.3	27.5	42.8	0.142	0.309	0.522	10.2	15.2	20.2	0.0	**6.2**	9.5	-2.8	**2.2**	120.0
4	**OCTOPUS-QJL**	2.2×	0.006	0.038	0.117	19.3	27.5	42.6	0.153	0.310	0.552	9.6	15.2	20.3	0.0	**6.2**	9.8	-4.0	**2.2**	120.0
3	TurboQuant-MSE	2.7×	0.013	0.098	0.263	15.5	22.6	35.5	0.207	0.423	0.661	8.5	13.1	18.8	0.0	6.5	10.0	-2.3	1.6	120.0
3	TurboQuant-QJL	2.7×	0.076	0.262	0.449	12.8	17.8	28.0	0.624	0.779	0.910	5.9	8.4	11.2	9.6	12.7	16.2	-19.5	-5.4	-0.1
3	PolarQuant	2.8×	0.013	0.093	0.263	16.1	22.8	34.9	0.218	0.402	0.691	9.0	13.1	19.5	0.0	**6.3**	10.1	-2.9	**1.7**	120.0
3	**OCTOPUS**	2.7×	0.012	**0.078**	0.228	16.5	**23.7**	36.0	0.196	0.390	0.628	8.4	**13.5**	20.8	0.0	6.4	10.1	-2.6	1.5	120.0
3	**OCTOPUS-QJL**	2.5×	0.012	**0.078**	0.225	16.5	**23.7**	36.4	0.207	**0.389**	0.624	8.8	**13.5**	20.4	0.0	6.4	10.1	-3.0	1.6	120.0
2	TurboQuant-MSE	3.2×	0.055	0.261	0.450	12.9	17.9	28.1	0.616	0.777	0.907	5.9	8.4	11.6	10.2	12.7	15.7	-17.6	-5.3	0.3
2	TurboQuant-QJL	3.2×	0.265	0.579	0.830	9.5	13.1	19.8	0.708	0.816	0.997	5.5	7.1	8.2	10.5	13.2	16.5	-28.9	-6.0	-0.2
2	PolarQuant	3.3×	0.045	0.251	0.469	13.0	17.9	28.3	0.575	0.727	0.898	5.5	8.6	11.2	9.9	12.6	16.0	-21.3	-5.3	2.3
2	**OCTOPUS**	3.1×	0.029	**0.178**	0.366	13.8	**19.7**	30.9	0.341	**0.581**	0.821	6.8	**10.9**	15.0	0.0	**6.8**	13.6	-9.6	**1.1**	120.0
2	**OCTOPUS-QJL**	2.8×	0.028	**0.178**	0.364	13.9	**19.7**	30.9	0.377	**0.581**	0.815	6.7	**10.9**	15.0	0.0	6.9	14.4	-8.2	1.0	120.0

12.6–13.2 dB mean LSD with negative mean SNR, while OCTOPUS remains at 6.75 dB LSD and +1.07 dB SNR. Even PolarQuant, the strongest non-OCTOPUS baseline, degrades 1.4× faster than OCTOPUS on mean LPIPS as bits decrease (CausVid: 0.25 vs. 0.18). Stills from the videos are provided in App. K.

4.4 Cross-modality patterns

All four modalities show the same pattern. **(i) OCTOPUS matches or beats every rotation baseline in the low-bit regimes where compression quality matters most.** Exceptions: $b{=}4$ video (Polar within 3%) and AAR at $b{=}3$, where PolarQuant is slightly better on mean LSD/SNR under 10 s AudioSet conditioning. **(ii) Competing codecs degrade catastrophically below $b{=}4$:** TurboQuant-QJL collapses to perceptual noise on CF at $b{=}2$ (max LPIPS ≈ 1.0); PolarQuant's worst prompt at $b{=}2$ hits LPIPS 0.90. OCTOPUS's worst prompt stays at 0.82—still degraded, but coherent. The extra b_{dir} bit (Eq. 10) provides a disproportionate MSE reduction at tight budgets. **(iii) QJL buys IP accuracy, not reconstruction-path quality;** OCTOPUS-QJL fits only score-attention deployments (Table 6).

Limitations. The improved rate-quality point is not free in wall-clock time: OCTOPUS adds more arithmetic than scalar Lloyd-Max decoding and remains slower than a bf16 SDPA path, so it is most attractive when KV bandwidth or capacity is the bottleneck (App. G).

5 Conclusion

OCTOPUS is a rotation-preconditioned KV codec that quantizes the rotated unit direction in contiguous triplets: an octahedral map [5, 10] collapses each 3-coordinate block to a pair of scalars on $[-1, 1]^2$, and Lloyd-Max [28, 29] quantizers matched to the norm and oct-coordinate marginals reduce the triplet to three integers under the asymmetric $(b{+}1, b{-}1)$ bit split. The codec inherits the data-oblivious online guaranties of TurboQuant [43] and combines without modification with the 1-bit QJL [42] residual. Across text, video, and audio, it matches or beats prior rotation codecs, with a lead that grows as bits drop. At $b{=}2$, OCTOPUS is often the only codec that does not collapse in long-context recall, and the only codec that retains usable perceptual quality in autoregressive video.

References

[1] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4895–4901, 2023.

[2] 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. *arXiv preprint*, 2024.

[3] Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, et al. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. *arXiv preprint*, 2024.

[4] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M. De Sa. QuIP: 2-bit quantization of large language models with guarantees. *Neural Information Processing Systems (NeurIPS)*, 36:4396–4429, 2023.

[5] Zina H. Cigolle, Sam Donow, Daniel Evangelakos, Michael Mara, Morgan McGuire, and Quirin Meyer. A survey of efficient representations for independent unit vectors. *Journal of Computer Graphics Techniques (JCGT)*, 3(2):1–30, 2014.

[6] Tri Dao, Daniel Y. Fu, Stefano Ermon, A. Rudra, and Christopher R'e. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Neural Information Processing Systems (NeurIPS)*, 2022.

[7] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. *Neural Information Processing Systems (NeurIPS)*, 35: 30318–30332, 2022.

[8] Shichen Dong, Wenfang Cheng, Jiayu Qin, and Wei Wang. QAQ: Quality adaptive quantization for LLM KV cache. *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2024.

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint*, 2024.

[10] Thomas Engelhardt and Carsten Dachsbacher. Octahedron environment maps. In *International Symposium on Vision, Modeling, and Visualization (VMV)*, 2008.

[11] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint*, 2022.

[12] Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128K context. *arXiv preprint*, 2024.

[13] Allen Gersho. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, 25(4):373–380, 1979.

[14] Allen Gersho. On the structure of vector quantizers. *IEEE Transactions on Information Theory*, 28(2):157–166, 1982.

[15] Insu Han, Praneeth Kacham, Amin Karbasi, Vahab Mirrokni, and Amir Zandieh. PolarQuant: Quantizing KV caches with polar transformation. *arXiv preprint*, 2025. Not to be confused with Wu et al. (arXiv:2502.00527), which shares the name "PolarQuant" but proposes a different method.

[16] Insu Han, Michael Kapralov, Ekaterina Kochetkova, Kshiteej Sheth, and Amir Zandieh. BalanceKV: KV cache compression through discrepancy theory. *arXiv preprint*, 2025.

[17] 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. *arXiv preprint*, 2024.

[18] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Proceedings of the Conference on Language Modeling (COLM)*, 2024.

[19] Greg Kamradt. Needle in a haystack — pressure testing LLMs. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`, 2023.

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

[21] Arseny Kapoulkine. Quantizing tangent frames. Blog post, `https://zeux.io/2026/04/30/quantizing-tangent-frames/`, 2026. Accessed 2026-04-30.

[22] Junhyuck Kim, Jongho Park, Jaewoong Cho, and Dimitris Papailiopoulos. Lexico: Extreme KV cache compression via sparse coding over universal dictionaries. *arXiv preprint*, 2024.

[23] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. *arXiv preprint*, 2024.

[24] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024.

[25] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics (TACL)*, 12:157–173, 2024.

[26] Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. *Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[27] Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2-bit quantization for KV cache. *arXiv preprint*, 2024.

[28] Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

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

[30] Philip F. Panter and Ward Dite. Quantization distortion in pulse-count modulation with nonuniform spacing of levels. *Proceedings of the IRE*, 39(1):44–48, 1951.

[31] Kai Qiu, Xiang Li, Hao Chen, Jie Sun, Jinglu Wang, Zhe Lin, Marios Savvides, and Bhiksha Raj. Efficient autoregressive audio modeling via next-scale prediction. *arXiv preprint*, 2024.

[32] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv preprint*, 2024.

[33] Zunhai Su, Zhe Chen, Wang Shen, Hanyu Wei, Linge Li, Huangqi Yu, and Kehong Yuan. RotateKV: Accurate and robust 2-bit KV cache quantization for LLMs via outlier-aware adaptive rotations. *arXiv preprint*, 2025.

[34] Philippe Tillet, H. T. Kung, and David Cox. Triton: An intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, 2019.

[35] Songhao Wu, Ang Lv, Xiao Feng, Yufei Zhang, Xun Zhang, Guojun Yin, Wei Lin, and Rui Yan. PolarQuant: Leveraging polar transformation for efficient key cache quantization and decoding acceleration. *arXiv preprint*, 2025.

[36] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, pages 38087–38099, 2023.

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

[38] June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. No token left behind: Reliable KV cache compression via importance-aware mixed precision quantization. *arXiv preprint*, 2024.

[39] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[40] Yuxuan Yue, Zhihang Yuan, Haojie Duanmu, Sifan Zhou, Jianlong Wu, and Liqiang Nie. WKVQuant: Quantizing weight and key/value cache for large language models gains more. *arXiv preprint*, 2024.

[41] Paul L. Zador. *Development and Evaluation of Procedures for Quantizing Multivariate Distributions*. PhD thesis, Stanford University, 1964.

[42] Amir Zandieh, Majid Daliri, and Insu Han. QJL: 1-bit quantized JL transform for KV cache quantization with zero overhead. *arXiv preprint*, 2024.

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

[44] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[45] Tianyi Zhang, Jonah Yi, Zhaozhuo Xu, and Anshumali Shrivastava. KV cache is 1 bit per channel: Efficient large language model inference with coupled quantization. *arXiv preprint*, 2024.

[46] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. *Neural Information Processing Systems (NeurIPS)*, 36, 2024.

[47] Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate LLM serving. In *Proceedings of Machine Learning and Systems*, pages 196–209, 2024.

[48] Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint*, 2026.

A Encoder and decoder algorithms

Algorithm 1 gives the encoder as it is implemented: one pass per key, with all intermediate state (rotated vector, triplet norms, octahedral coordinates, integer indices) kept in registers. At decode time, Algorithm 2 fuses bit unpacking, octahedral decode, centroid gather, value dequantization, and online softmax into a single split-K flash-decoding kernel in the style of Dao et al. [6], Shah et al. [32]. Keys are reconstructed in registers on the fly, and the only per-step memory traffic is the packed KV state and the running softmax statistics (m, ℓ, acc). A reduction kernel merges the per-split triples with the standard flash-decoding identity. These kernels are an artefact of the implementation, not of the algorithm: every number in Section 4 can be reproduced with a pure PyTorch reference of the same mathematical operations at proportionally higher wall-clock cost.

Algorithm 1 OCTOPUS encoder with joint (ξ, η, ρ) rounding (Sec. 3.5), one program per key. Line 14 is the 3×3 optimal-rounding refinement; setting the candidate set Δ to $\{(0,0)\}$ recovers the legacy scalar-rounding baseline.

Require: $\boldsymbol{k} \in \mathbb{R}^d$, sign vector $\boldsymbol{s} \in \{\pm1\}^d$, codebook centroids $\mathcal{C}_\xi \in \mathbb{R}^{2^{b_{\mathrm{dir}}}}$, $\mathcal{C}_\rho \in \mathbb{R}^{2^{b_{\mathrm{nrm}}}}$, boundaries $\mathcal{B}_\xi, \mathcal{B}_\rho$ (midpoints of adjacent centroids), candidate set $\Delta \subseteq \{-1,0,1\}^2$ (default $\Delta = \{-1,0,1\}^2$).

1: $\gamma \leftarrow \sqrt{\sum_{i=1}^d k_i^2}$ ▷ fp32 norm
2: $\tilde{\boldsymbol{u}} \leftarrow \boldsymbol{k} / \max(\gamma, \epsilon)$
3: $\boldsymbol{u} \leftarrow \boldsymbol{H}\,(\boldsymbol{s} \odot \tilde{\boldsymbol{u}})$ ▷ in-register WHT butterfly with normalized $\boldsymbol{H}$
4: **for** $i = 0$ **to** $n_{\mathrm{tri}} - 1$ **do**
5: $\quad \boldsymbol{t}_i \leftarrow \boldsymbol{u}_{3i:3i+3}$; $\quad (x, y, z) \leftarrow \boldsymbol{t}_i$
6: $\quad (p_x, p_y, p_z) \leftarrow (x, y, z) / \max(|x| + |y| + |z|, \epsilon)$
7: $\quad$ **if** $p_z \geq 0$ **then**
8: $\quad\quad (\xi_i, \eta_i) \leftarrow (p_x, p_y)$
9: $\quad$ **else**
10: $\quad\quad (\xi_i, \eta_i) \leftarrow \big(\mathrm{sign}(p_x)(1 - |p_y|),\ \mathrm{sign}(p_y)(1 - |p_x|)\big)$
11: $\quad$ **end if**
12: $\quad j_x \leftarrow \mathrm{searchsorted}(\mathcal{B}_\xi, \xi_i)$; $j_y \leftarrow \mathrm{searchsorted}(\mathcal{B}_\xi, \eta_i)$ ▷ scalar seed
13: $\quad s^* \leftarrow -\infty$; $\quad (J_x, J_y) \leftarrow (j_x, j_y)$
14: $\quad$ **for** $(\delta_x, \delta_y) \in \Delta$ **do**
15: $\quad\quad j'_x \leftarrow \mathrm{clip}(j_x + \delta_x, 0, 2^{b_{\mathrm{dir}}} - 1)$; $j'_y \leftarrow \mathrm{clip}(j_y + \delta_y, 0, 2^{b_{\mathrm{dir}}} - 1)$
16: $\quad\quad s \leftarrow \boldsymbol{t}_i^\top\, \mathrm{Oct}^{-1}(\mathcal{C}_\xi[j'_x], \mathcal{C}_\xi[j'_y])$ ▷ Eq. 6
17: $\quad\quad$ **if** $s > s^*$ **then** $(s^*, J_x, J_y) \leftarrow (s, j'_x, j'_y)$
18: $\quad$ **end for**
19: $\quad I^{\mathrm{dir}}_{i,0} \leftarrow J_x$; $I^{\mathrm{dir}}_{i,1} \leftarrow J_y$
20: $\quad I^{\mathrm{nrm}}_i \leftarrow \mathrm{searchsorted}(\mathcal{B}_\rho,\ \mathrm{clip}(s^*, 0, 1))$
21: **end for**
22: **return** $\big(\gamma,\ \mathrm{pack}(\{I^{\mathrm{dir}}\},\ b_{\mathrm{dir}}),\ \mathrm{pack}(\{I^{\mathrm{nrm}}\},\ b_{\mathrm{nrm}})\big)$

Algorithm 2 OCTOPUS split-K flash decode, one program per (head, split).

Require: $\boldsymbol{q}_{\mathrm{rot}} \in \mathbb{R}^d$, $(\gamma_t, \mathcal{I}_{\mathrm{dir},t}, \mathcal{I}_{\mathrm{nrm},t})_{t \in \mathrm{chunk}}$, V codec state, codebook centroids $\boldsymbol{c}_\xi$, $\boldsymbol{c}_\rho$.

1: $(m, \ell, \mathrm{acc}) \leftarrow (-\infty,\ 0,\ \mathbf{0})$
2: **for** t **in** chunk **do**
3: $\quad$ Load packed bytes $\mathcal{I}_{\mathrm{dir},t}, \mathcal{I}_{\mathrm{nrm},t}$ and γ_t.
4: $\quad$ Extract $(I^{\mathrm{dir}}_{t,i,0}, I^{\mathrm{dir}}_{t,i,1}, I^{\mathrm{nrm}}_{t,i})_{i=0}^{n_{\mathrm{tri}}-1}$ via shift+mask.
5: $\quad$ Gather $(\hat{\xi}_{t,i}, \hat{\eta}_{t,i}, \hat{\rho}_{t,i})$ from $(\boldsymbol{c}_\xi, \boldsymbol{c}_\rho)$.
6: $\quad \hat{\boldsymbol{n}}_{t,i} \leftarrow \mathrm{Oct}^{-1}(\hat{\xi}_{t,i}, \hat{\eta}_{t,i})$.
7: $\quad s_t \leftarrow \gamma_t \cdot \sum_i \hat{\rho}_{t,i}\, \boldsymbol{q}_{\mathrm{rot},i}^\top \hat{\boldsymbol{n}}_{t,i}$.
8: $\quad (m, \ell, \mathrm{acc}) \leftarrow \mathrm{onlineSoftmax}(m, \ell, \mathrm{acc}, s_t/\sqrt{d}, \hat{\boldsymbol{v}}_t)$.
9: **end for**
10: Emit partial (m, ℓ, acc) to split buffer.

B Mathematical details

Proposition 1 (Inner-product invariance)**.** *For any $\boldsymbol{q}, \boldsymbol{k} \in \mathbb{R}^d$ and $\boldsymbol{s} \in \{\pm 1\}^d$, $\boldsymbol{q}^\top \boldsymbol{k} = (\boldsymbol{R}\boldsymbol{q})^\top (\boldsymbol{R}\boldsymbol{k}) = \gamma\, (\boldsymbol{R}\boldsymbol{q})^\top \boldsymbol{u}$.*

Any quantization of $\boldsymbol{u}$ at decode time can therefore be combined with a matching rotation of the query, and the dot product is unbiased in expectation.

Proposition 2 (Marginal concentration [43, Thm. 1])**.** *If $\tilde{\boldsymbol{u}}$ is uniformly distributed on S^{d-1} and $\boldsymbol{s}$ is independent uniform on $\{\pm 1\}^d$, each coordinate u_i of $\boldsymbol{u} = \boldsymbol{R}\tilde{\boldsymbol{u}}$ has the symmetric-Beta density of Eq. 3.*

Proof of theorem 3.1. Let $\boldsymbol{g} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ and set $\boldsymbol{u} = \boldsymbol{g}/\|\boldsymbol{g}\|_2$. Then $\boldsymbol{u}$ is uniform on S^{d-1}. Since the random-signed WHT $\boldsymbol{R}$ is orthogonal, it preserves this distribution, so the rotated direction $\boldsymbol{R}\boldsymbol{u}$ has the same law as $\boldsymbol{u}$. It therefore suffices to consider any full three-coordinate block of $\boldsymbol{u}$.

For such a triplet,

$$\rho_i^2 = u_{3i+1}^2 + u_{3i+2}^2 + u_{3i+3}^2 = \frac{A}{A+B},$$

where

$$A = \sum_{j=3i+1}^{3i+3} g_j^2 \sim \chi_3^2, \qquad B = \sum_{j \notin \{3i+1, 3i+2, 3i+3\}} g_j^2 \sim \chi_{d-3}^2.$$

The variables A and B are independent because they are sums over disjoint Gaussian coordinates. Since $\chi_k^2 = \mathrm{Gamma}(k/2, 2)$, the standard gamma-ratio identity gives

$$\rho_i^2 = \frac{A}{A+B} \sim \mathrm{Beta}\left(\frac{3}{2}, \frac{d-3}{2}\right).$$

Finally, applying the change of variables $x = \rho_i^2$, $dx = 2\rho_i\, d\rho_i$, yields

$$f_\rho(r) = \frac{2r^2(1-r^2)^{(d-5)/2}}{B\left(\frac{3}{2}, \frac{d-3}{2}\right)}, \qquad r \in [0, 1].$$

□

C Derivations

C.1 Magnitude-Direction Split (Equation 1)

For any nonzero key $\boldsymbol{k} \in \mathbb{R}^d$, set

$$\gamma = \|\boldsymbol{k}\|_2, \qquad \tilde{\boldsymbol{u}} = \boldsymbol{k}/\gamma.$$

Then

$$\|\tilde{\boldsymbol{u}}\|_2 = \frac{\|\boldsymbol{k}\|_2}{\gamma} = 1,$$

so $\tilde{\boldsymbol{u}} \in S^{d-1}$ and $\boldsymbol{k} = \gamma\tilde{\boldsymbol{u}}$. The implementation stores the original norm γ and divides by a clamped positive denominator only to handle zero or tiny keys safely.

C.2 Sign-Flipped WHT Rotation (Equation 2)

Let $\boldsymbol{D}_{\boldsymbol{s}} = \mathrm{diag}(\boldsymbol{s})$ with $s_i \in \{\pm 1\}$, and let $\boldsymbol{H}$ be the normalized Hadamard matrix. The implemented rotation is

$$\boldsymbol{R} = \boldsymbol{H}\boldsymbol{D}_{\boldsymbol{s}}, \qquad \boldsymbol{u} = \boldsymbol{R}\tilde{\boldsymbol{u}}.$$

Because $\boldsymbol{H}^\top \boldsymbol{H} = \boldsymbol{I}$ and $\boldsymbol{D}_{\boldsymbol{s}}^\top \boldsymbol{D}_{\boldsymbol{s}} = \boldsymbol{I}$,

$$\boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{D}_{\boldsymbol{s}}^\top \boldsymbol{H}^\top \boldsymbol{H}\boldsymbol{D}_{\boldsymbol{s}} = \boldsymbol{I}.$$

Thus $\boldsymbol{R}$ maps S^{d-1} to itself. Since the normalized Hadamard is self-inverse, $\boldsymbol{R}^{-1} = \boldsymbol{R}^\top = \boldsymbol{D}_{\boldsymbol{s}}\boldsymbol{H}$. The same identity gives the rotated-frame inner product

$$\boldsymbol{q}^\top \boldsymbol{k} = \gamma\, \boldsymbol{q}^\top \boldsymbol{R}^\top \boldsymbol{u} = \gamma\, (\boldsymbol{R}\boldsymbol{q})^\top \boldsymbol{u}.$$

C.3 Coordinate Marginal (Equation 3)

For $\boldsymbol{u} \sim \mathrm{Unif}(S^{d-1})$, one coordinate U satisfies $(U+1)/2 \sim \mathrm{Beta}(a,a)$ with $a = (d-1)/2$. With $z = (u+1)/2$ and $dz/du = 1/2$,

$$\begin{aligned} f_U(u) &= \frac{1}{2B(a,a)} \left(\frac{1+u}{2}\right)^{a-1} \left(\frac{1-u}{2}\right)^{a-1} \\ &= \frac{(1-u^2)^{(d-3)/2}}{B((d-1)/2,(d-1)/2)\ 2^{d-2}}, \qquad u \in [-1,1]. \end{aligned}$$

This is the density implemented by the scalar MSE Lloyd-Max codebook. For OCTOPUS, it mainly justifies the rotated-sphere prior; the default codec uses the triplet and octahedral marginals below.

C.4 Triplet Split (Section 3.2)

Pad $\boldsymbol{u}$ with zeros to length $3n_{\mathrm{tri}}$ and split the padded vector into contiguous triplets $\boldsymbol{t}_i$. Since padding adds only zeros,

$$\sum_i \|\boldsymbol{t}_i\|_2^2 = \|\boldsymbol{u}\|_2^2 = 1.$$

Therefore $0 \le \rho_i = \|\boldsymbol{t}_i\|_2 \le 1$. If $\rho_i > 0$, then

$$\boldsymbol{n}_i = \boldsymbol{t}_i/\rho_i, \qquad \|\boldsymbol{n}_i\|_2 = 1,$$

so $\boldsymbol{n}_i \in S^2$ and $\boldsymbol{t}_i = \rho_i \boldsymbol{n}_i$. When $\rho_i = 0$, the direction is mathematically arbitrary; the reference and Triton encoders use an ϵ-safe divisor to produce a finite placeholder.

C.5 Triplet-Norm Marginal (Equation 4)

Generate a uniform sphere point as $\boldsymbol{u} = \boldsymbol{z}/\|\boldsymbol{z}\|_2$ with $z_j \overset{\mathrm{iid}}{\sim} \mathcal{N}(0,1)$. For any three-coordinate triplet,

$$\rho_i^2 = \frac{z_1^2+z_2^2+z_3^2}{\sum_{j=1}^d z_j^2} = \frac{X}{X+Y},$$

where $X \sim \chi_3^2$, $Y \sim \chi_{d-3}^2$, and X, Y are independent. Hence

$$\rho_i^2 \sim \mathrm{Beta}\left(\frac{3}{2}, \frac{d-3}{2}\right).$$

Let $S = \rho_i^2$ and $R = \rho_i$. The change of variables $s = r^2$ gives $ds/dr = 2r$, so

$$\begin{aligned} f_R(r) &= f_S(r^2)\, 2r \\ &= \frac{2r(r^2)^{1/2}(1-r^2)^{(d-5)/2}}{B(3/2,(d-3)/2)} \\ &= \frac{2r^2(1-r^2)^{(d-5)/2}}{B(3/2,(d-3)/2)}, \qquad r \in [0,1]. \end{aligned}$$

This is the density integrated by the implemented triplet-norm Lloyd-Max codebook.

C.6 Octahedral Encode (Equation 5)

For $\boldsymbol{n} = (x,y,z) \in S^2$, define $\ell = |x| + |y| + |z|$ and $\boldsymbol{p} = \boldsymbol{n}/\ell$. Then $|p_x| + |p_y| + |p_z| = 1$, so $\boldsymbol{p}$ lies on the unit L_1 octahedron. If $p_z \ge 0$, the upper face is already parameterized by (p_x, p_y). If $p_z < 0$, the lower face is folded into the square by

$$(\xi,\eta) = (\mathrm{sign}(p_x)(1-|p_y|), \mathrm{sign}(p_y)(1-|p_x|)).$$

On the fold boundary $p_z = 0$, the two branches agree because $|p_x| + |p_y| = 1$. The implementation uses the convention $\mathrm{sign}(0) = +1$ and clamps denominators away from zero.

C.7 Octahedral Decode (Equation 6)

Given $(\xi, \eta) \in [-1, 1]^2$, set $r = 1 - |\xi| - |\eta|$. If $r \geq 0$, the point is on an unfolded upper face and the unnormalized vector is (ξ, η, r). If $r < 0$, inverting the fold gives

$$\xi' = \operatorname{sign}(\xi)(1 - |\eta|), \qquad \eta' = \operatorname{sign}(\eta)(1 - |\xi|).$$

Thus, with $(\xi', \eta') = (\xi, \eta)$ in the $r \geq 0$ branch,

$$\boldsymbol{n}(\xi, \eta) = \frac{(\xi', \eta', r)}{\|(\xi', \eta', r)\|_2}.$$

The final normalization maps the octahedral surface back to S^2; both the reference decoder and the fused attention kernel implement this formula on dequantized oct-coordinate centroids.

C.8 Octahedral Marginal (Equation 7)

For the implemented fold, the induced density on the square is obtained by composing the affine octahedral inverse with radial projection onto S^2. On each triangular fold region, write the unnormalized inverse as $\boldsymbol{q}(\xi, \eta)$. Uniform surface measure pulls back to

$$g(\xi, \eta) = \frac{1}{4\pi} \|\boldsymbol{q}(\xi, \eta)\|_2^{-3},$$

because each affine face has unit volume factor in $|\det(\boldsymbol{q}, \partial_\xi \boldsymbol{q}, \partial_\eta \boldsymbol{q})|$. By symmetry the marginal depends only on $a = |\xi|$. Integrating the two inner and two folded vertical segments gives

$$f_\xi(\xi) = \frac{1}{2\pi} \left[\int_0^{1-a} \frac{ds}{(a^2 + s^2 + (1 - a - s)^2)^{3/2}} + \int_0^a \frac{ds}{(s^2 + (1 - a)^2 + (s - a)^2)^{3/2}} \right],$$

and evaluating the integrals yields

$$f_\xi(\xi) = \frac{1}{\pi \sqrt{a^2 + (1 - a)^2}} \left(\frac{1 - a}{1 - 2a + 3a^2} + \frac{a}{2 - 4a + 3a^2} \right), \qquad a = |\xi|.$$

The implementation does not evaluate this expression directly; it samples $\boldsymbol{n} \sim \mathrm{Unif}(S^2)$, applies the same octahedral fold, and trains a one-dimensional Lloyd-Max codebook on the empirical coordinate marginal.

C.9 Triplet MSE Bound (Equation 8)

With $\boldsymbol{t}_i = \rho_i \boldsymbol{n}_i$ and $\hat{\boldsymbol{t}}_i = \hat{\rho}_i \hat{\boldsymbol{n}}_i$, add and subtract $\rho_i \hat{\boldsymbol{n}}_i$:

$$\begin{aligned} \|\boldsymbol{t}_i - \hat{\boldsymbol{t}}_i\|_2^2 &= \|\rho_i(\boldsymbol{n}_i - \hat{\boldsymbol{n}}_i) + (\rho_i - \hat{\rho}_i)\hat{\boldsymbol{n}}_i\|_2^2 \\ &\leq 2\rho_i^2 \|\boldsymbol{n}_i - \hat{\boldsymbol{n}}_i\|_2^2 + 2(\rho_i - \hat{\rho}_i)^2 \|\hat{\boldsymbol{n}}_i\|_2^2 \\ &= 2(\rho_i - \hat{\rho}_i)^2 + 2\rho_i^2 \|\boldsymbol{n}_i - \hat{\boldsymbol{n}}_i\|_2^2. \end{aligned}$$

The last line uses $\|\hat{\boldsymbol{n}}_i\|_2 = 1$, enforced by octahedral decode normalization.

C.10 Expected MSE Budget (Equation 9)

Under the Gaussian sphere construction above, the block radius and direction are independent: the Gaussian block direction is uniform on S^2, while the block and complement radii determine ρ_i. Also

$$\mathbb{E}[\rho_i^2] = \frac{3/2}{3/2 + (d - 3)/2} = \frac{3}{d}.$$

Taking expectations in the triplet bound and using high-rate scalar Lloyd-Max/Panter-Dite scaling,

$$\mathbb{E}[(\rho_i - \hat{\rho}_i)^2] \approx C_\rho \sigma_\rho^2 4^{-b_{\mathrm{nrm}}}, \qquad \mathbb{E}[\|\boldsymbol{n}_i - \hat{\boldsymbol{n}}_i\|_2^2] \approx C_n \sigma_{\boldsymbol{n}}^2 4^{-b_{\mathrm{dir}}},$$

gives

$$\begin{aligned} \mathbb{E}\big[\|\boldsymbol{t}_i - \hat{\boldsymbol{t}}_i\|_2^2\big] &\approx 2C_\rho \sigma_\rho^2 4^{-b_{\mathrm{nrm}}} + 2\mathbb{E}[\rho_i^2] C_n \sigma_{\boldsymbol{n}}^2 4^{-b_{\mathrm{dir}}} \\ &= 2C_\rho \sigma_\rho^2 4^{-b_{\mathrm{nrm}}} + (6/d) C_n \sigma_{\boldsymbol{n}}^2 4^{-b_{\mathrm{dir}}}. \end{aligned}$$

Here σ_ρ^2 denotes the variance of the scalar norm ρ, which is what the implementation quantizes.

C.11 Lagrangian Bit Allocation (Equation 10)

Let

$$A = 2C_\rho \sigma_\rho^2, \qquad D = \frac{6}{d} C_n \sigma_{\boldsymbol{n}}^2.$$

The high-rate objective is

$$f(b_{\mathrm{nrm}}, b_{\mathrm{dir}}) = A4^{-b_{\mathrm{nrm}}} + D4^{-b_{\mathrm{dir}}}, \qquad b_{\mathrm{nrm}} + 2b_{\mathrm{dir}} = B_{\mathrm{tri}}.$$

For the Lagrangian:

$$\mathcal{L} = A4^{-b_{\mathrm{nrm}}} + D4^{-b_{\mathrm{dir}}} + \lambda(b_{\mathrm{nrm}} + 2b_{\mathrm{dir}} - B_{\mathrm{tri}}),$$

stationarity (deriving by b_{nrm} and b_{dir} and equating to 0) gives

$$-A \log_4 4^{-b_{\mathrm{nrm}}} + \lambda = 0, \qquad -D \log_4 4^{-b_{\mathrm{dir}}} + 2\lambda = 0.$$

Thus

$$A4^{-b_{\mathrm{nrm}}} = \frac{D}{2} 4^{-b_{\mathrm{dir}}},$$

and

$$b_{\mathrm{dir}}^\star - b_{\mathrm{nrm}}^\star = \log_4\left(\frac{D}{2A}\right) = \log_4\left(\frac{3C_n\sigma_{\boldsymbol{n}}^2}{2dC_\rho\sigma_\rho^2}\right).$$

The factor of two in the denominator is the shadow price of spending one additional bit on each of two octahedral coordinates.

C.12 Bit-Gap Scaling

The squared triplet norm has $\mathrm{Var}(\rho_i^2) = \mathcal{O}(d^{-2})$, but the implemented norm codebook quantizes ρ_i itself. Since $\rho_i = \chi_3/\sqrt{d} + o(d^{-1/2})$ under the sphere prior,

$$\sigma_\rho^2 = \mathrm{Var}(\rho_i) = \mathcal{O}(d^{-1}), \qquad \sigma_{\boldsymbol{n}}^2 = \mathcal{O}(1).$$

Substituting these scalings into the corrected stationarity condition gives

$$b_{\mathrm{dir}}^\star - b_{\mathrm{nrm}}^\star = \mathcal{O}(1),$$

up to constants from the source densities and the octahedral metric. The implemented $(b+1, b-1)$ split is therefore a finite-dimensional codebook choice supported by the empirical sweeps, not a consequence of a growing asymptotic gap for direct ρ quantization.

C.13 Joint Rounding (Equation 12)

$$\ell(\hat{\xi}_i, \hat{\eta}_i, \hat{\rho}_i) = ||\boldsymbol{t}_i - \hat{\rho}_i \boldsymbol{n}(\hat{\xi}_i, \hat{\eta}_i)||_2^2 \tag{14}$$

$$= \rho_i^2 ||\boldsymbol{n}(\xi_i, \eta_i)||_2^2 - 2\hat{\rho}_i \boldsymbol{t}_i^\top \boldsymbol{n}(\hat{\xi}_i, \hat{\eta}_i) + \hat{\rho}_i^2 ||\boldsymbol{n}(\hat{\xi}_i, \hat{\eta}_i)||_2^2 \tag{15}$$

$$= \rho_i^2 - 2\hat{\rho}_i s_i(\xi_i, \eta_i) + \hat{\rho}_i^2 \tag{16}$$

where $||\boldsymbol{n}(\xi_i, \eta_i)||_2^2 = \boldsymbol{n}(\hat{\xi}_i, \hat{\eta}_i)||_2^2 = 1$, and $s_i(\hat{\xi}_i, \hat{\eta}_i) = \boldsymbol{t}_i^\top \boldsymbol{n}(\hat{\xi}_i, \hat{\eta}_i)$ i.e. the dot product of the true vector and the quantized direction candidate. This means minimum of this parabola on $\hat{\rho}_i$ occurs exactly when $\hat{\rho}_i = s_i$. Therefore, the best quantized radius is NOT the centroid nearest to the true radius (ρ_i), but the centroid nearest to the projected dot product (s_i).

C.14 Score Factorization (Equation 13)

The decoded key has $\hat{\boldsymbol{k}} = \gamma \boldsymbol{R}^\top \hat{\boldsymbol{u}}$, so

$$\boldsymbol{q}^\top \hat{\boldsymbol{k}} = \gamma\, (\boldsymbol{R}\boldsymbol{q})^\top \hat{\boldsymbol{u}} = \gamma\, \boldsymbol{q}_{\mathrm{rot}}^\top \hat{\boldsymbol{u}}.$$

The decoder reconstructs each rotated triplet as $\hat{\boldsymbol{u}}_i = \hat{\rho}_i \hat{\boldsymbol{n}}_i$, where

$$\hat{\boldsymbol{n}}_i = \mathrm{Oct}^{-1}\left(\mathcal{C}_\xi[I_{i,0}^{\mathrm{dir}}], \mathcal{C}_\xi[I_{i,1}^{\mathrm{dir}}]\right), \qquad \hat{\rho}_i = \mathcal{C}_\rho[I_i^{\mathrm{nrm}}].$$

Therefore

$$\boldsymbol{q}^\top \hat{\boldsymbol{k}} = \gamma \sum_{i=0}^{n_{\mathrm{tri}}-1} \hat{\rho}_i\, \boldsymbol{q}_{\mathrm{rot},i}^\top \hat{\boldsymbol{n}}_i.$$

The fused attention kernel applies the usual attention scale after this raw score is formed.

Table 4: **Diagonal bit-split sweep $(b{+}\delta, b{-}\delta)$, relative to the uniform (b, b) reference.** Synthetic Gaussian keys, $d{=}128$, $n{=}8192$, 4 seeds. "Δ MSE" and "$\Delta(1-\cos)$" are the percentage change against the uniform (b, b) baseline at the same b; negative means the off-diagonal split improves on uniform. "invalid" marks diagonal endpoints with a zero-bit component (no codebook). The implemented split sits at $\delta{=}{+}1$ for every $b \in \{2, 3, 4\}$ and is the unique diagonal step that reduces MSE versus uniform.

b	δ	b_{dir}	b_{nrm}	MSE	$1-\cos$	Δ MSE vs (b,b)	$\Delta(1-\cos)$ vs (b,b)
2	-1	1	3	0.4555	0.2628	$+223.3\%$	$+260.3\%$
2	0	2	2	0.1409	0.0729	0.0% (ref)	0.0% (ref)
2	**+1**	**3**	**1**	**0.0831**	**0.0421**	$\mathbf{-41.0\%}$	$\mathbf{-42.4\%}$
3	-2	1	5	0.4501	0.2593	$+1104.8\%$	$+1279.8\%$
3	-1	2	4	0.1273	0.0658	$+240.6\%$	$+250.2\%$
3	0	3	3	0.0374	0.0188	0.0% (ref)	0.0% (ref)
3	**+1**	**4**	**2**	**0.0243**	**0.0120**	$\mathbf{-35.0\%}$	$\mathbf{-36.1\%}$
3	$+2$	5	1	0.0537	0.0268	$+43.8\%$	$+42.8\%$
4	-2	2	6	0.1262	0.0653	$+1217.1\%$	$+1265.3\%$
4	-1	3	5	0.0332	0.0168	$+246.8\%$	$+250.5\%$
4	0	4	4	0.0096	0.0048	0.0% (ref)	0.0% (ref)
4	**+1**	**5**	**3**	**0.0067**	**0.0033**	$\mathbf{-30.5\%}$	$\mathbf{-31.7\%}$
4	$+2$	6	2	0.0166	0.0081	$+72.9\%$	$+69.7\%$

C.15 QJL Residual Estimator

Let $\boldsymbol{r} = \boldsymbol{u} - \hat{\boldsymbol{u}}$ and $\gamma_r = \|\boldsymbol{r}\|_2$. For an independent ideal QJL projection $\boldsymbol{R}'$, store

$$\boldsymbol{\sigma} = \operatorname{sign}(\boldsymbol{R}'\boldsymbol{r}) \in \{\pm 1\}^d.$$

For $a = \boldsymbol{q}_{\text{rot}}$, the asymmetric one-bit estimator is

$$\hat{z}(a, \boldsymbol{r}) = \sqrt{\frac{\pi}{2d}}\, \gamma_r\, (\boldsymbol{R}'a)^\top \operatorname{sign}(\boldsymbol{R}'\boldsymbol{r}).$$

Under the ideal QJL model,

$$\mathbb{E}\big[(\boldsymbol{R}'a)_j \operatorname{sign}((\boldsymbol{R}'\boldsymbol{r})_j)\big] = \sqrt{\frac{2}{\pi d}}\, \frac{a^\top \boldsymbol{r}}{\gamma_r}.$$

Summing over d coordinates and multiplying by $\sqrt{\pi/(2d)}\,\gamma_r$ gives

$$\mathbb{E}[\hat{z}(a, \boldsymbol{r})] = a^\top \boldsymbol{r}.$$

Since $\boldsymbol{q}^\top \boldsymbol{k} = \gamma\, \boldsymbol{q}_{\text{rot}}^\top(\hat{\boldsymbol{u}} + \boldsymbol{r})$, the corrected score is

$$\boldsymbol{q}^\top \hat{\boldsymbol{k}} + \gamma\, \hat{z}(\boldsymbol{q}_{\text{rot}}, \boldsymbol{r}).$$

The implementation uses the same scaling with a structured WHT projection and stores γ_r in fp16, so exact unbiasedness is the ideal-model statement.

D Bit-allocation sweep

We sweep the diagonal $(b{+}\delta, b{-}\delta)$, $\delta \in \{-2, -1, 0, +1, +2\}$, around each uniform reference $b \in \{2, 3, 4\}$ on $n{=}8192$ random Gaussian keys at $d{=}128$, averaged over 4 rotation seeds. Table 4 is the diagonal sweep with every entry expressed relative to the uniform (b, b) baseline.

E Rounding ablation

Table 5 reports the four rounding modes supported by the reference encoder (Sec. 3.5) at matched bits. All modes share the same bitstream and decoder, so this is a pure encoder ablation. Metrics are averaged over five seeds on $n{=}4096$ random Gaussian keys at $d{=}128$ with $n_{\text{query}}{=}64$. *tail95* is

Table 5: **OCTOPUS rounding-mode ablation.** Scalar rounding vs. the joint-rounding variants of Sec. 3.5. "Δ tail95" is the percentage change in the 95th-percentile squared error relative to the scalar baseline. The 3×3 local search is byte-identical to the full direction search at every bit width tested.

b	rounding	cos ↑	MSE ↓	tail95 ↓	|ip err| ↓	Δ MSE	Δ tail95
1	scalar	0.692	0.6022	0.7935	7.065	–	–
1	local_2×2	0.703	0.5172	0.6664	6.554	−14.1%	−16.0%
1	**local_3×3**	**0.703**	**0.5172**	**0.6664**	**6.554**	**−14.1%**	**−16.0%**
1	full	0.703	0.5172	0.6664	6.554	−14.1%	−16.0%
2	scalar	0.955	0.0897	0.1205	2.722	–	–
2	local_2×2	0.957	0.0858	0.1152	2.662	−4.4%	−4.3%
2	**local_3×3**	**0.958**	**0.0832**	**0.1119**	**2.620**	**−7.2%**	**−7.1%**
2	full	0.958	0.0832	0.1119	2.620	−7.2%	−7.1%
3	scalar	0.987	0.0261	0.0365	1.464	–	–
3	local_2×2	0.988	0.0250	0.0351	1.433	−4.0%	−3.7%
3	**local_3×3**	**0.988**	**0.0243**	**0.0343**	**1.414**	**−6.6%**	**−5.9%**
3	full	0.988	0.0243	0.0343	1.414	−6.6%	−5.9%
4	scalar	0.997	0.0071	0.0102	0.763	–	–
4	local_2×2	0.997	0.0068	0.0098	0.749	−3.8%	−3.6%
4	**local_3×3**	**0.997**	**0.0067**	**0.0096**	**0.739**	**−6.1%**	**−5.7%**
4	full	0.997	0.0067	0.0096	0.739	−6.1%	−5.7%

the 95th-percentile per-key squared reconstruction error. The full-codebook direction search is the joint-optimum upper bound reachable with this $(\mathcal{C}_\xi, \mathcal{C}_\rho)$ pair; `local_3x3` is the implemented default of Algorithm 1.

Takeaways consistent with the tangent-frame survey of Kapoulkine [21]: optimal rounding mostly shifts the encoder, not the codebook; the gain is largest at the tightest bit budgets (where one misrounding consumes a large fraction of the remaining precision); and a small local neighborhood suffices in practice. The implemented 3×3 search gives a uniform 6–14% MSE reduction at matched bit rate with *zero* change to the bitstream format or the decoder. Because only the encoder is affected, previously serialised scalar-rounded states remain valid; the joint-rounding improvement applies to every new OCTOPUS state produced under the default encoder, including the OCTOPUS-QJL variant whose residual stage sees a correspondingly smaller Stage-A error.

F QJL effective-rate accounting

Table 6 spells out the effective-rate cost of adding the one-bit residual side-car described in Sec. 3.6.

G Kernel speed and KV compression

Table 7 reports wall-clock encode and decode times on a single NVIDIA H200 (B=1, median of 50 runs after 30 warm-up iterations) at each modality's full sequence length. The SDPA bf16 baseline uses pre-cast bf16 KV tensors for decode (no per-step cast overhead); its "encode" column reports the one-time fp32→bf16 copy cost.

The fused decode kernels add 5–11× overhead vs. the highly optimised cuDNN SDPA bf16 path, decreasing at lower bit widths as the packed data shrinks. This overhead is inherent: each decode step fuses decompression—centroid lookup (TQ-MSE) or octahedral reconstruction (OCTOPUS)—into the attention loop, trading compute for the KV memory savings reported in the last column. TQ-MSE encode is lightweight (≤2 ms even at 65k tokens); OCTOPUS encode uses a Kronecker-factored WHT and direct triad indexing, bringing the per-token cost to ≈0.08 μs (d_h=128), which is negligible for auto-regressive LLM decode and small relative to diffusion denoising for video.

Table 6: **QJL effective-rate accounting.** Effective bits per KV scalar (analytically computed from the codec configuration: nominal K/V bits + g=32/16 group metadata + 1-bit JL residual + scale storage) compared at matched nominal bits. OCTOPUS-QJL adds a 1-bit JL residual on top of OCTOPUS, raising the effective rate by exactly 0.5 bits per scalar and giving up roughly that much reconstruction headroom inside the standard *dequantize-then-dot* attention path. We therefore recommend the QJL variant only for score-attention deployments where the residual is consumed by a custom kernel.

modality	nominal b	bits/scalar	metric	QJL bits/scalar	QJL metric
LLM (WikiText-2↓)	4	4.50	10.306	5.00	10.306
	3	3.50	10.753	4.00	10.754
	2	2.50	13.517	3.00	13.511
CausVid (LPIPS↓)	4	6.92	0.038	7.40	0.038
	3	5.98	0.078	6.46	0.078
	2	5.15	0.178	5.63	0.178
Causal Forcing (LPIPS↓)	4	5.91	0.309	6.45	0.310
	3	4.87	0.390	5.40	0.389
	2	3.95	0.581	4.48	0.581
AAR (LSD dB↓)	4	7.24	6.20	7.59	6.17
	3	6.58	6.45	6.93	6.43
	2	6.06	6.75	6.40	6.88

H Long-context needle-in-a-haystack sweep

Table 8 expands the long-context retrieval summary from Sec. 4.2 across the full context-length and bit-width grid.

I Memory-budget Pareto

Figure 4 recasts the LLM results as a deployment memory trade-off, making the Pareto frontier visible at fixed context length.

J Full per-modality tables

Table 9, Table 10, and Table 11 provide the per-codec metrics behind the modality summary in Sec. 4.3.

K Stills

Figure 5 shows representative worst-case frames for the video codecs, complementing the aggregate LPIPS/PSNR/SSIM numbers in Sec. 4.3.

Table 7: **Encode / decode kernel speed and KV compression.** Per decode step on a single NVIDIA H200. "Encode" is the Triton compression time for T key-value tokens; "Decode" is a single fused-attention query against T compressed tokens. SDPA bf16 encode is the bf16 KV copy baseline. OCTOPUS encode uses the 3×3 joint rounding search (Sec. 3.5). KV ratio = bf16 bytes / compressed bytes.

b	codec	encode (ms)	decode (ms)	dec / base	KV ratio
Video (Wan-1.3B, H=16, d_h=64, g=32, T=32,760)					
–	SDPA bf16	0.11	0.06	1.0×	1.0×
4	OCTOPUS	3.8	0.59	10.4×	2.8×
4	TQ-MSE	2.0	0.48	8.5×	3.0×
3	OCTOPUS	3.6	0.50	8.9×	3.6×
3	TQ-MSE	2.1	0.45	7.9×	3.8×
2	OCTOPUS	3.6	0.49	8.6×	4.5×
2	TQ-MSE	2.0	0.34	6.0×	4.9×
Audio (AAR, H=16, d_h=64, g=16, T=455)					
–	SDPA bf16	0.02	0.03	1.0×	1.0×
4	OCTOPUS	0.10	0.15	5.9×	2.4×
4	TQ-MSE	0.19	0.15	5.6×	2.6×
3	OCTOPUS	0.10	0.16	5.9×	2.9×
3	TQ-MSE	0.24	0.15	5.6×	3.0×
2	OCTOPUS	0.10	0.16	6.0×	3.5×
2	TQ-MSE	0.19	0.15	5.8×	3.8×
LLM (Qwen2.5-7B GQA, H_{kv}=4, d_h=128, g=32, T=65,536)					
–	SDPA bf16	0.11	0.06	1.0×	1.0×
4	OCTOPUS	5.2	0.66	11.3×	3.0×
4	TQ-MSE	1.5	0.37	6.4×	3.1×
3	OCTOPUS	4.6	0.55	9.4×	3.7×
3	TQ-MSE	1.6	0.33	5.7×	3.9×
2	OCTOPUS	4.5	0.52	8.9×	4.8×
2	TQ-MSE	1.5	0.28	4.9×	5.1×

Table 8: **Multi-key needle-in-a-haystack recall on `Qwen2.5-7B-Instruct-1M`.** RULER-style multi-key protocol: each cell plants four distractor needles plus one target needle, each carrying a fresh random 8-character alphanumeric magic value; scoring is exact-match on the target value. Recall is averaged over 5 depth offsets $\{0.0, 0.25, 0.5, 0.75, 1.0\}$ per context length. Higher is better. Per column, the best rank is **bold** (every tied codec); the runner-up is underlined only when the column has at least three distinct values, and a saturated column (every codec at the same recall) is left unhighlighted.

bits	codec	4k	8k	16k	32k	64k	128k
–	fp16 baseline	1.00	1.00	1.00	1.00	1.00	1.00
4	TurboQuant-MSE	1.00	**1.00**	1.00	1.00	1.00	1.00
4	TurboQuant-QJL	1.00	**1.00**	1.00	1.00	1.00	1.00
4	PolarQuant	1.00	**1.00**	1.00	1.00	1.00	1.00
4	**OCTOPUS**	1.00	**1.00**	1.00	1.00	1.00	1.00
4	**OCTOPUS-QJL**	1.00	0.95	1.00	1.00	1.00	1.00
3	TurboQuant-MSE	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
3	TurboQuant-QJL	0.80	0.65	0.75	0.60	0.55	0.65
3	PolarQuant	0.90	0.75	0.75	0.90	**1.00**	0.85
3	**OCTOPUS**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
3	**OCTOPUS-QJL**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
2	TurboQuant-MSE	**0.85**	0.70	0.60	0.50	0.65	0.55
2	TurboQuant-QJL	0.05	0.00	0.00	0.00	0.00	0.00
2	PolarQuant	0.05	0.05	0.05	0.05	0.05	0.00
2	**OCTOPUS**	**0.85**	**0.90**	**0.80**	0.75	0.85	0.70
2	**OCTOPUS-QJL**	0.80	0.85	**0.80**	**0.85**	**0.90**	**0.75**

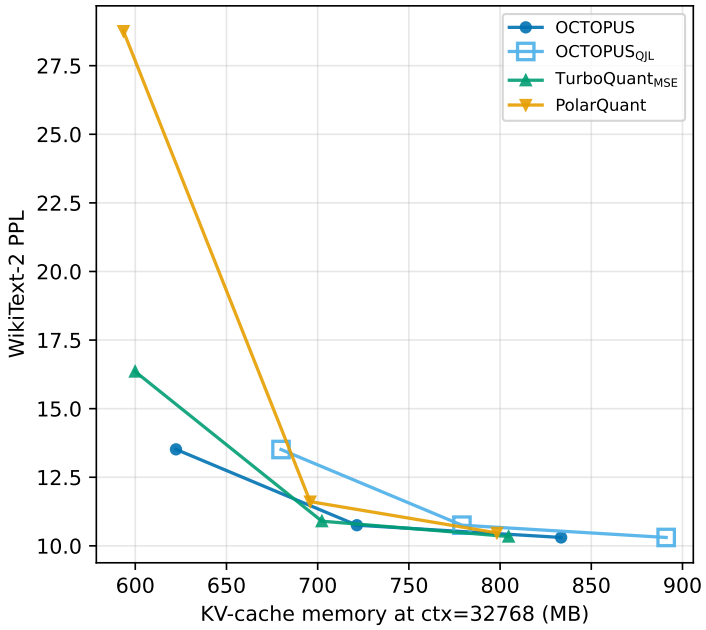


Figure 4: **LLM quality at fixed deployment memory.** WikiText-2 perplexity vs. KV-cache memory at 32,768-token context for `Qwen2.5-7B-Instruct-1M`. The probe-time `kv_cache_bytes` is linearly extrapolated from the sweep's measurement window to 32k tokens, so the x-axis is the memory budget a deployment actually pays. OCTOPUS dominates the Pareto frontier across the full memory range; OCTOPUS-QJL trails slightly because the 1-bit JL residual costs a constant $\sim$60 MB but does not move the reconstruction-quality curve at this context length.

Table 9: **CausVid video generation, expanded.** Default recipe (residual window $=$ 1 frame, V group g=32, no per-layer boundary protection). All five codecs at the same recipe so only the codec varies. Higher is better for PSNR/SSIM/CLIP/latent-cos; lower for LPIPS. Best per bit width is **bold**, runner-up underlined.

bits	codec	compr.	LPIPS ↓	PSNR ↑	SSIM ↑	CLIP ↑	lat-cos ↑
4	TurboQuant-MSE	2.40×	0.0454	26.47	0.8807	0.3168	0.9894
4	TurboQuant-QJL	2.38×	0.0963	22.62	0.8051	**0.3175**	0.9680
4	PolarQuant	2.42×	**0.0369**	**27.82**	**0.8984**	0.3171	**0.9922**
4	**OCTOPUS**	2.31×	0.0380	27.52	0.8945	**0.3175**	0.9920
4	**OCTOPUS-QJL**	2.16×	0.0380	27.52	0.8946	0.3174	**0.9922**
3	TurboQuant-MSE	2.75×	0.0975	22.56	0.8029	0.3172	0.9674
3	TurboQuant-QJL	2.72×	0.2621	17.81	0.6402	**0.3204**	0.8707
3	PolarQuant	2.77×	0.0928	22.84	0.8111	0.3168	0.9682
3	**OCTOPUS**	2.67×	**0.0777**	**23.72**	**0.8298**	0.3172	**0.9754**
3	**OCTOPUS-QJL**	2.48×	0.0778	23.71	0.8297	0.3173	**0.9754**
2	TurboQuant-MSE	3.22×	0.2611	17.87	0.6429	**0.3211**	0.8717
2	TurboQuant-QJL	3.19×	0.5790	13.10	0.4503	0.3067	0.5110
2	PolarQuant	3.26×	0.2514	17.93	0.6618	0.3171	0.8728
2	**OCTOPUS**	3.11×	0.1784	**19.68**	0.7190	0.3180	0.9225
2	**OCTOPUS-QJL**	2.84×	**0.1783**	**19.68**	**0.7192**	0.3179	**0.9226**

Table 10: **Causal Forcing video generation, expanded.** Same recipe and conventions as Table 9.

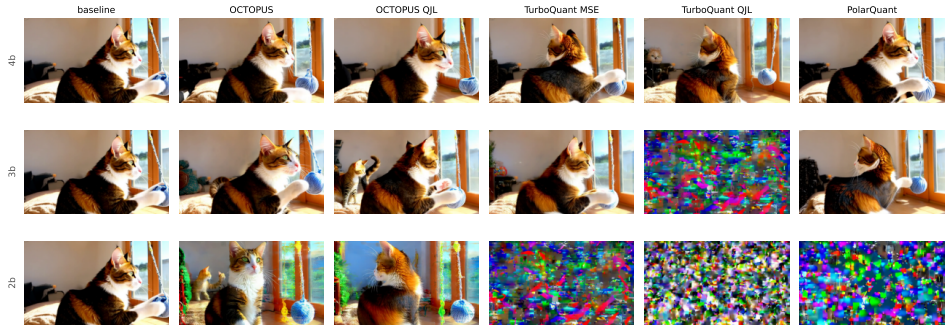
bits	codec	compr.	LPIPS ↓	PSNR ↑	SSIM ↑	CLIP ↑	lat-cos ↑
4	TurboQuant-MSE	2.84×	0.3339	14.58	0.5547	<u>0.3163</u>	0.8051
4	TurboQuant-QJL	2.81×	0.4213	13.09	0.4884	**0.3169**	0.7374
4	PolarQuant	2.87×	**0.3009**	**15.41**	**0.5850**	0.3156	**0.8275**
4	**OCTOPUS**	2.71×	<u>0.3093</u>	15.21	0.5757	0.3152	<u>0.8192</u>
4	**OCTOPUS-QJL**	2.48×	0.3103	<u>15.23</u>	<u>0.5775</u>	0.3162	0.8190
3	TurboQuant-MSE	3.41×	0.4225	13.10	0.4861	**0.3174**	0.7347
3	TurboQuant-QJL	3.37×	0.7786	8.42	0.1664	0.1513	0.2497
3	PolarQuant	3.46×	0.4018	13.06	0.4971	0.3144	0.7471
3	**OCTOPUS**	3.29×	<u>0.3904</u>	<u>13.48</u>	<u>0.5090</u>	0.3157	<u>0.7594</u>
3	**OCTOPUS-QJL**	2.96×	**0.3892**	**13.54**	**0.5123**	<u>0.3158</u>	**0.7627**
2	TurboQuant-MSE	4.28×	0.7770	8.45	0.1664	0.1499	0.2508
2	TurboQuant-QJL	4.21×	0.8164	7.10	0.1106	0.0845	0.1481
2	PolarQuant	4.35×	<u>0.7273</u>	8.62	0.2155	0.2089	0.2984
2	**OCTOPUS**	4.05×	**0.5808**	**10.90**	**0.3593**	<u>0.3024</u>	**0.5618**
2	**OCTOPUS-QJL**	3.57×	**0.5808**	<u>10.89</u>	<u>0.3577</u>	**0.3034**	<u>0.5592</u>

Table 11: **Autoregressive audio (AAR), expanded.** Metrics averaged over 100 random 10 s AudioSet-20k clips used as CLAP-audio conditioning at the default recipe (residual window 1 scale, V group g=16, no per-layer protection). Best per bit width is **bold**, runner-up <u>underlined</u>.

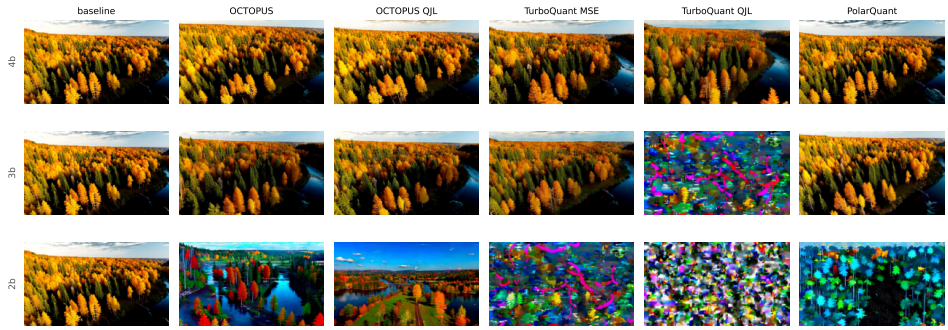
bits	codec	compr.	LSD ↓	mel-MSE ↓	SNR ↑	lat-cos ↑
4	TurboQuant-MSE	2.29×	6.36	0.2191	2.07	–
4	TurboQuant-QJL	2.26×	6.24	0.2284	1.80	–
4	PolarQuant	2.31×	6.28	<u>0.2130</u>	**2.23**	–
4	**OCTOPUS**	2.21×	<u>6.20</u>	0.2134	<u>2.22</u>	–
4	**OCTOPUS-QJL**	2.11×	**6.17**	**0.2104**	2.19	–
3	TurboQuant-MSE	2.48×	6.48	0.2377	1.55	–
3	TurboQuant-QJL	2.46×	12.72	1.4883	-5.44	–
3	PolarQuant	2.51×	**6.34**	**0.2279**	**1.71**	–
3	**OCTOPUS**	2.43×	6.45	<u>0.2343</u>	1.51	–
3	**OCTOPUS-QJL**	2.31×	<u>6.43</u>	0.2346	<u>1.60</u>	–
2	TurboQuant-MSE	2.72×	12.65	1.4547	-5.30	–
2	TurboQuant-QJL	2.69×	13.17	1.6676	-5.98	–
2	PolarQuant	2.75×	12.59	1.4325	-5.28	–
2	**OCTOPUS**	2.64×	**6.75**	**0.3171**	**1.07**	–
2	**OCTOPUS-QJL**	2.50×	<u>6.88</u>	<u>0.3240</u>	<u>0.97</u>	–

(a) CausVid

(b) CausVid



(c) Causal Forcing



(d) Causal Forcing

Figure 5: **Worst-case codec divergence across bit depths.** Each panel shows the single frame with the highest combined cross-codec L1 divergence from the fp16 baseline (same frame index for both pipelines). Rows: $b{=}4, 3, 2$; columns: baseline and each codec. OCTOPUS remains visually faithful at every bit width; competing codecs collapse at $b{\leq}3$.