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 independent of the total dimensionality of the keys, reducing MSE over balanced scalar Lloyd-Max for every model family 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 all but a few operating points, with a lead that grows with more extreme compression. Furthermore, a fused Triton implementation reconstructs keys on the fly without materializing the uncompressed key, with a small constant per-step overhead over scalar dequantization. End-to-end decode on Qwen2.5-7B is 1.11–$1.16\times$ faster than TurboQuant-MSE at matched bit width. Code and project page: `https://octopus-quant.github.io/`.

1 Introduction

Long-context autoregressive inference, such as in large language models (LLM) [11], causal video generation models [44, 53], or audio generation models [34], is dominated by reading the key-value (KV) cache from high-bandwidth memory at every decoding step [14, 28]. KV compression is therefore the primary target for both latency and batch-size optimization, and prior works address it through token eviction [26, 41, 51], per-channel scalar quantization with residuals [19, 23, 30], and more recently rotation-preconditioned quantization codecs [17, 47, 48].

Rotation-based codecs depend on a structured random orthogonal $\boldsymbol{R}$ (typically a sign-flipped Walsh-Hadamard transform due to efficiency [6]) to make the marginal of every rotated key coordinate isotropic and *analytically* known. A 1-D Lloyd-Max quantizer [31, 32] matched to that marginal is then near-optimal at matched bit width. In this way, TurboQuant [48] gets a symmetric Beta marginal, PolarQuant [17] does the analogous construction on recursive polar angles, and the QJL 1-bit residual makes the dot product unbiased at near-zero memory cost [47]. 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

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

non-uniform bit allocation between them, beats the per-coordinate quantizers at matched rate. *Second*, the octahedral map from computer graphics [7, 12] is a low-distortion 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. This is what lets the same codec run unchanged on language, video, and audio decoders, where a calibrated vector quantizer would need a fresh codebook per model, layer distribution, and modality. Our contributions are:

- **Octahedral triplet direction quantizer** as a KV cache primitive, with implementation-matched norm and direction marginals. Holding everything else fixed, the octahedral encoding outperforms spherical angles, raw Cartesian coordinates, and a calibrated 2-D codebook, and matches a larger learned spherical codebook at a fraction of its size. The compress-decode pipeline is implemented as fused Triton kernels [8, 35, 37] that reconstruct keys on the fly from packed bit indices and never needs to materialize the full key tensor (with only a small constant per-step overhead over scalar dequantization).
- **An MSE-optimal non-uniform bit split.** A Lagrangian on the per-triplet squared error yields a stationarity condition independent of d and total budget, supporting the implemented $(b{+}1, b{-}1)$ as the optimal integer split on both synthetic keys and keys from three LLM families.
- **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 [42]), chunk-wise video diffusion (CausVid [44]), frame-wise causal video forcing [53], and next-scale autoregressive audio [34], 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 [5, 26, 29, 41, 51] 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 [10, 19, 23, 30, 43, 45, 50]. Sparse coding [25] trades a bigger code table for ultra-low rates. Rotation-preconditioned codecs [2, 17, 18, 36, 39, 47, 48] 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 [48] 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 [31, 32] codebook depends only on (d, b) and lands within a small constant of the Zador-Gersho [15, 46] bound. The structured Walsh-Hadamard transform with random sign flips is the standard fast preconditioner [2, 6, 36]. PolarQuant [17] parameterises the rotated direction recursively in polar coordinates instead. We share only the Walsh-Hadamard preconditioner with these codecs, which is the standard choice of essentially every rotation-based KV codec. The downstream stages differ: OCTOPUS quantizes blocks of three rotated coordinates jointly via an octahedral direction+norm split with a Lagrangian bit allocation and joint rounding, which we show gives strictly lower MSE than per-coordinate scalar quantization 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 [7, 12]; 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 [47] shows that a 1-bit Johnson-Lindenstrauss sketch gives an unbiased inner-product estimator at essentially zero memory; we compose it with

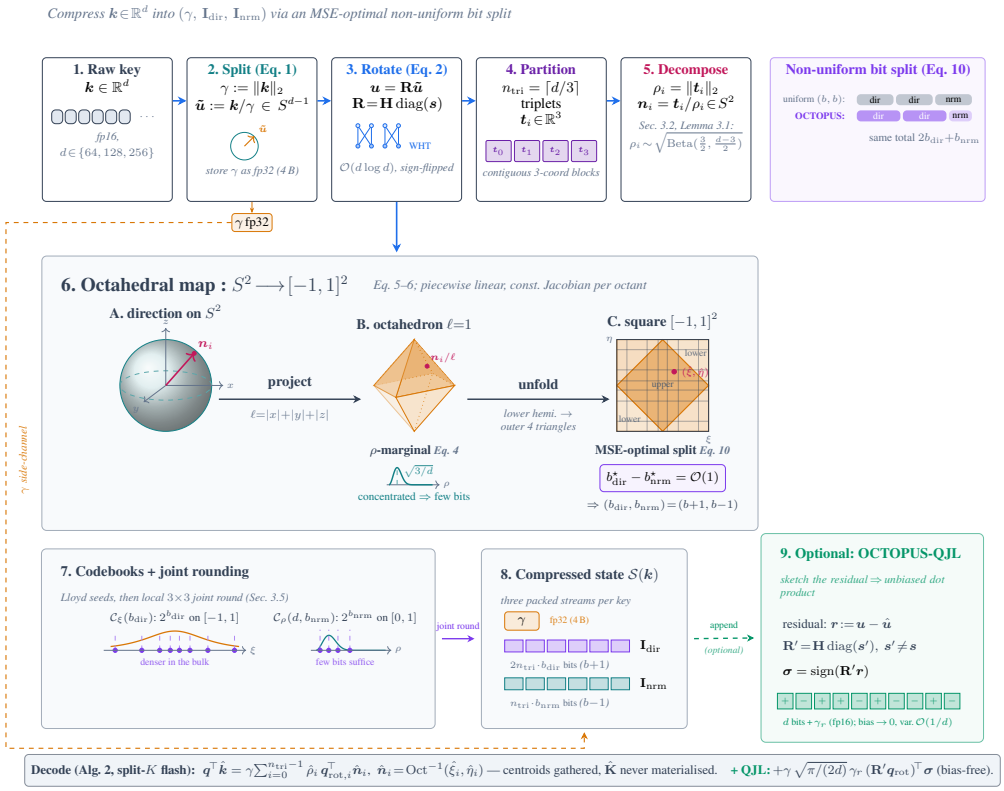


Figure 1: **The OCTOPUS encode pipeline.** Stages 1–5 (top) realise the rotation and triplet decomposition of Sec. 3.1–3.2: a key $\boldsymbol{k}$ is normalised (Eq. 1), preconditioned by a sign-flipped Walsh-Hadamard rotation (Eq. 2), cut into $n_{\text{tri}} = \lceil d/3 \rceil$ triplets, and decomposed 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).

OCTOPUS under the tag OCTOPUS-QJL. We borrow only the rotation idea from the broader quantization literature on weights [6, 13, 27] and weight+activation quantization [2, 9, 40, 52]; the codec, bit allocation and codebooks are specific to the KV cache and online by construction. Fused attention kernels [8, 35] keep our reconstruction in registers.

Vector quantization in a rotated frame. QuIP# [38] pairs a randomized Hadamard rotation with an E_8 lattice codebook for *weights*, establishing that joint quantization improves on scalar quantization in a rotated frame. QuIP# is an offline weight quantizer with Hessian-based calibration; OCTOPUS is the analogous construction for the online, data-oblivious, multimodal KV regime. A learned codebook would have to be refit for every decoder, layer distribution, or modality, and has no direct path into a fused attention kernel. We compare against a calibrated 3-D k-means VQ in Sec. 4.4.

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 $\tilde{\boldsymbol{u}} \in S^{d-1}$:

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

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,

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

$\boldsymbol{R}$ 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 [48, Thm. 1]. OCTOPUS instead quantizes *triplets* of rotated coordinates jointly. We partition $\boldsymbol{u}$ into $n_{\mathrm{tri}} = \lceil d/3 \rceil$ contiguous triplets $\boldsymbol{t}_i \in \mathbb{R}^3$. Standard head dimensions ($d \in \{64, 128, 256\}$) are not divisible by three, so the last one or two coordinates are zero-padded ("tail padding"); on Qwen2.5 keys this costs only 0.06% (B_{tri}=7) and 2.19% (B_{tri}=10) MSE relative to coding the incomplete block with a scalar quantizer at the same budget (App. G.4). The block size of three is a design choice rather than a geometric necessity: it is the dimension at which the octahedral map is the standard $S^2 \to [-1,1]^2$ encoding with an $\mathcal{O}(1)$ fused decode, and it aligns with the norm-plus-direction split below. 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}, $\rho_i^2 \sim \mathrm{Beta}(3/2, (d-3)/2)$, 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 [7, 12]. 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 $[-1,1]^2$:

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

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 $(\xi',\eta') = (\mathrm{sign}(\xi)(1-|\eta|), \mathrm{sign}(\eta)(1-|\xi|))$ otherwise. The map is a piecewise linear bijection $S^2 \to [-1,1]^2$ with a constant Jacobian per octant [7, 12]. The octahedral fold maps to a *square* code space, so per-coordinate Lloyd-Max on (ξ,η) is a tractable approximation of 2-sphere distortion, while recursive polar parameterizations [17] need transcendental operators and induce $\sin^{2^{\ell-1}-1}(2\psi)$ angle marginals. Two independent scalar quantizers on (ξ,η) are not equivalent to a true S^2 quantizer; the constant-per-octant Jacobian keeps the two induced distortions comparable, and the decoder's final normalization projects back onto S^2. Sec. 4.4 measures the net effect: holding everything else fixed, the octahedral encoding outperforms spherical angles, raw Cartesian coordinates, and a calibrated 2-D codebook, and matches a learned spherical direction codebook at a fraction of its size.

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

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

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 $\mathrm{Oct}(\boldsymbol{n})$, $\boldsymbol{n} \sim \mathrm{Unif}(S^2)$, and shares it between ξ and η.

3.3 MSE-optimal bit allocation

OCTOPUS quantizes each triplet in a total budget of $B_{\mathrm{tri}} = 2b_{\mathrm{dir}} + b_{\mathrm{nrm}}$ 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 $b_{\mathrm{dir}}{=}b_{\mathrm{nrm}}{=}b$ giving $B_{\mathrm{tri}}{=}3b$. 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

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

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 [16, 33], a 1-D Lloyd-Max quantizer with b bits and source variance σ^2 incurs $D \approx C\sigma^2 4^{-b}$. For both the triplet-norm and the oct-coordinate marginal, this law meets a 10% error-and-slope criterion only from 4 bits on, so the deployed components with fewer bits lie outside the strict high-rate regime; we use it only to obtain the *form* of the stationarity condition, and establish the finite-rate optimum empirically below. 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

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

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

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

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

Scope of the optimality claim. Eq. 10 is the squared-error optimum *within the structured direction-plus-norm codebook class* (one scalar norm codebook and one scalar codebook shared by the two octahedral coordinates) at any total triplet budget B_{tri}. It does not imply optimality over arbitrary 3-D codebooks: an unconstrained, data-calibrated 3-D vector quantizer can reach lower MSE at the same rate, at the cost of per-model calibration, a 38–154$\times$ larger codebook, and no direct path to a fused attention decode (Sec. 4.4).

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. At a fixed budget of $B = 3b$, the only non-degenerate integral option is $2b_{\mathrm{dir}}{+}b_{\mathrm{nrm}}{=}3b$. However, $B_{\mathrm{tri}}{=}3b{+}1$ yields $(b{+}1, b{-}1)$ as the best allocation (App. G), reducing MSE over balanced scalar Lloyd-Max at all nine tested points (Sec. 4.4).

3.4 Codebooks

Two Lloyd-Max codebooks suffice: $\mathcal{C}_\rho(d, b_{\text{nrm}})$ on $[0,1]$ matched to Eq. 4, and $\mathcal{C}_\xi(b_{\text{dir}})$ 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 $(d, b_{\text{dir}}, b_{\text{nrm}})$, 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 [24], 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 $\hat{\rho}_i = \arg\min_{c \in \mathcal{C}_\rho} |c - \text{clip}_{[0,1]}(s_i^*)|$. Direction selection therefore decouples from ρ selection.

Local 3×3 candidate set. The full direction argmax runs over $2^{2b_{\text{dir}}}$ 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 $b_{\text{dir}} \in \{2, \ldots, 5\}$, this search was *byte-identical* to the full grid search in all buckets at a fraction of the cost (App. E) and we adopt this heuristic search range going forward.

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:

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

where $\boldsymbol{q}_{\text{rot}} = \boldsymbol{R}\boldsymbol{q}$ and $\hat{\boldsymbol{n}}_i = \text{Oct}^{-1}(\mathcal{C}_\xi[I_{i,0}^{\text{dir}}], \mathcal{C}_\xi[I_{i,1}^{\text{dir}}])$, $\hat{\rho}_i = \mathcal{C}_\rho[I_i^{\text{nrm}}]$. Only $2n_{\text{tri}}$ 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 [48]. We optionally attach a QJL [47] sketch of the rotated-frame residual $\boldsymbol{r} \coloneqq \boldsymbol{u} - \hat{\boldsymbol{u}}$. With $\boldsymbol{R}' = \boldsymbol{H}\,\text{diag}(\boldsymbol{s}')$ a second rotation with independent seed, we store $\boldsymbol{\sigma} = \text{sign}(\boldsymbol{R}'\boldsymbol{r}) \in \{\pm 1\}^d$ and a residual norm $\gamma_r = \|\boldsymbol{r}\|_2$ (fp16). The QJL estimator $\widehat{\boldsymbol{q}_{\text{rot}}^\top \boldsymbol{r}} = \sqrt{\pi/(2d)}\,\gamma_r\,(\boldsymbol{R}'\boldsymbol{q}_{\text{rot}})^\top \boldsymbol{\sigma}$ 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 $\boldsymbol{q}^\top \hat{\boldsymbol{k}} + \gamma\, \widehat{\boldsymbol{q}_{\text{rot}}^\top \boldsymbol{r}}$.

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** [48]

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**

(per-coordinate Lloyd-Max), **TurboQuant-QJL** [47, 48] (MSE stage + 1-bit JL residual), and **PolarQuant** [17] (recursive polar). The only variable across rows is the K codec, and every comparison is matched at the same nominal symmetric $K{=}V$ bit width b. The deployed OCTOPUS key format spends one extra bit per triplet ($3b{+}1$ vs. $3b$), which is reflected in the KV× columns; Sec. 4.4 reports exactly rate-matched comparisons.

Modalities. (i) Synthetic isotropic Gaussian keys at $d{=}128$, where the rotation-Beta-Lloyd baseline is provably near-optimal; (ii) `Qwen2.5-7B-Instruct-1M` [42] (GQA [1], 28 layers, $d_h{=}128$); (iii) two Wan-1.3B autoregressive video DiTs ($d_h{=}64$); (iv) the next-scale audio model AAR [34]. Recipes are given per modality below; compression ratios are fp16 over 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 [48]). 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 (App. F): OCTOPUS 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. [48, §5] we report WikiText-2 and C4 perplexity (PPL) (512-token blocks, 8 chunks) and a multi-key needle-in-a-haystack sweep [20, 22] (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.

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. J). At $b{=}4$ all codecs reach 1.00. At $b{=}3$, OCTOPUS holds 1.00; PolarQuant drops to 0.86

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$\times$ = 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×

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.

Downstream long-context tasks. On a 7-task LongBench subset [3] (first 20 test examples each of NarrativeQA, Qasper, MultiFieldQA-en, HotpotQA, 2WikiMQA, GovReport, and QMSum; middle truncation to 31,500 tokens; greedy decoding of up to 128 tokens; official LongBench-v1 scoring) at $b{=}3$ with the boundary-1 recipe, OCTOPUS scores 15.61 vs. 14.68 for TurboQuant-MSE and 16.85 for bf16 (App. K).

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 [44], which generates in 3-frame chunks, and Causal Forcing [53], 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 [49], PSNR, SSIM, and latent cosine against the uncompressed rollout.

The audio experiment uses AAR [34], 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 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 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. N.

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

4.4 Codec-level ablations on real keys

Because OCTOPUS is post-training and calibration-free, its design choices can be evaluated in isolation on offline, rotated keys. We dump the keys of every layer and KV head of Qwen2.5-7B-Instruct-1M [42], SmolLM3-3B [4], and Mistral-7B-v0.3 [21] ($d{=}128$) on four 256-token prompts, split them by (prompt, layer) into a calibration set and a disjoint held-out set (1,024 evaluation keys, 64 queries), and report reconstruction MSE (per coordinate, on the raw keys, so absolute values are not comparable across models) at per-triplet budgets $B_{\text{tri}} \in \{7, 10, 13\}$, i.e. the deployed $(b{+}1, b{-}1)$ formats at $b \in \{2, 3, 4\}$.

Exact-rate scalar baseline. At exactly the same $B_{\text{tri}}{=}3b{+}1$, OCTOPUS reduces MSE over balanced per-coordinate Lloyd-Max at all nine points (Table 4, top): by 3.5–8.1% on SmolLM3 and Mistral and by 12.9–28.5% on Qwen2.5, where the gap widens as bits drop. Within the octahedral class, $(b{+}1, b{-}1)$ is also the lowest-MSE integer allocation at $B_{\text{tri}}{=}3b{+}1$ on all three families; at $B_{\text{tri}}{=}3b$ the integer constraint only admits gaps $b_{\text{dir}}{-}b_{\text{nrm}} \in \{0, 3\}$ and the uniform split is best (App. G).

Calibrated 3-D vector quantization. As an upper reference we fit a generalized-Lloyd (k-means) 3-D VQ with $2^{B_{\text{tri}}}$ centroids, one global codebook per (model, rate) shared across layers and heads, on 65,536 rotated triplets. OCTOPUS beats it on Qwen2.5 at $B_{\text{tri}}{=}7$ (5.3% lower MSE); elsewhere the VQ is 18–24% lower in MSE, but requires per-model calibration and a 38–154× larger codebook (1.5/12 KiB vs. 40/80 B), and its irregular centroid lookup has no direct path into a fused attention decode. OCTOPUS thus trades some MSE against a calibrated VQ for being online, data-oblivious, and fusible—the properties that let it run unchanged across the modalities of Sec. 4.3.

Direction encoding. Holding triplet decomposition, bit allocation, norm codebook, and total budget fixed, we swap only the direction encoding (Table 4, bottom). On all three families the octahedral map beats spherical angles, raw Cartesian coordinates, and a calibrated VQ on the octahedral pair (+3–85% MSE for the alternatives). An unstructured spherical direction VQ with $2^{2b_{\text{dir}}}$ centroids is the only competitive option: 2.1–4.9% worse on Qwen2.5 and within ±1% on SmolLM3 and Mistral, but with a 10–38× larger codebook and no fused decode path. The octahedral map is thus the best *structured* direction encoding we tested, matching a learned spherical codebook at a fraction of its cost. Absolute numbers for all families, head-dimension sensitivity, and the combinability of OCTOPUS with other cache techniques are in App. G.

4.5 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

Table 4: **Codec-level ablations on real rotated keys.** Reconstruction MSE on held-out triplets; B_{tri} is the per-triplet bit budget ($3b{+}1$ for the deployed $(b{+}1, b{-}1)$ split). *Top:* relative MSE of OCTOPUS vs. balanced scalar Lloyd-Max at the same rate, and of a calibrated 3-D k-means VQ vs. OCTOPUS (negative: VQ better). *Bottom:* Qwen2.5 per-coordinate MSE on raw keys when only the direction encoding is swapped (calibrated alternatives); relative change vs. OCTOPUS in parentheses. Other families: App. G.

	OCTOPUS vs. scalar Lloyd-Max			3-D VQ vs. OCTOPUS	
model	$B_{\mathrm{tri}}{=}7$	10	13	7	10
Qwen2.5	-28.5%	-17.4%	-12.9%	$+5.3\%$	-18.3%
SmolLM3	-4.7%	-3.5%	-5.2%	-19.9%	-23.9%
Mistral	-5.4%	-5.9%	-8.1%	-19.7%	-22.9%

B_{tri}	**OCTOPUS**	sph. angles	raw Cartesian	oct-pair VQ	sph. VQ
7	**7.775**	10.453 (+34.4%)	11.400 (+46.6%)	9.643 (+24.0%)	8.158 (+4.9%)
10	**2.882**	3.006 (+4.3%)	5.332 (+85.1%)	2.978 (+3.3%)	2.943 (+2.1%)

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 13).

Limitations. The improved rate-quality point is not free in compute. In isolation, the fused OCTOPUS decode kernel is 1.05–$1.45\times$ slower per step than the TurboQuant-MSE kernel across modalities (1.10–$1.37\times$ on Qwen2.5-7B), owing to the octahedral fold. In our end-to-end Qwen2.5-7B benchmark OCTOPUS decodes 1.11–$1.16\times$ faster than TurboQuant-MSE, but at batch size 1 both codecs reach only 0.24–$0.34\times$ of bf16 SDPA throughput (App. I), so OCTOPUS is most attractive when KV bandwidth or capacity is the bottleneck. A per-model calibrated 3-D vector quantizer attains lower MSE than OCTOPUS at most tested rates (Sec. 4.4), at the price of calibration and of losing the fused decode path. The block size is fixed to three, so head dimensions not divisible by three rely on tail padding, and the fused kernel is currently compiled for $d \leq 256$.

5 Conclusion

OCTOPUS is a rotation-preconditioned KV codec that quantizes the rotated unit direction in contiguous triplets: an octahedral map [7, 12] collapses each 3-coordinate block to a pair of scalars on $[-1, 1]^2$, and Lloyd-Max [31, 32] 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 guarantees of TurboQuant [48] and combines without modification with the 1-bit QJL [47] 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. Its optimality is scoped to the structured direction-plus-norm codebook class: a calibrated 3-D VQ can reach lower MSE on a single model, but forgoes the calibration-free, fused, multimodal deployment that OCTOPUS provides.

Broader impact. By lowering the memory cost of autoregressive inference, OCTOPUS enables slightly more efficient, slimmed-down deployment of existing models across modalities, and we hope it draws further attention to efficiency research. It adds no new model capabilities in and of itself.

Acknowledgments and Disclosure of Funding

This work was carried out as part of the authors' employment at Stability AI. None of the authors received third-party funding or support for this work and the authors had no competing interests beyond this employment.

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] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3119–3137, 2024.

[4] Elie Bakouch, Carlos Miguel Patiño, Anton Lozhkov, Edward Beeching, Aymeric Roucher, Nouamane Tazi, et al. SmolLM3: Smol, multilingual, long-context reasoner. Hugging Face blog, `https://huggingface.co/blog/smollm3`, 2025.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

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

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

[17] 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.

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

[19] 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.

[20] 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.

[21] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint*, 2023.

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

[23] 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.

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

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with Hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning (ICML)*, pages 48630–48656, 2024.

[39] 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.

[40] 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.

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

[42] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, et al. Qwen2.5-1M technical report. *arXiv preprint*, 2025. arXiv:2501.15383.

[43] 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.

[44] 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.

[45] 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.

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

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

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

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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. [8], Shah et al. [35]. 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 [48, 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 5: **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). Each row spends $B_{\text{tri}}{=}3b{+}\delta$ bits, so this sweep is not iso-rate; the exactly rate-matched comparison is in App. G. 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

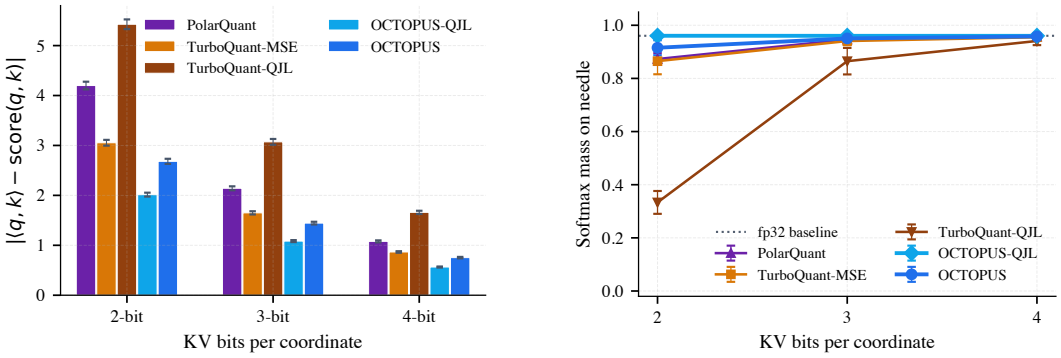
We sweep the diagonal $(b{+}\delta, b{-}\delta)$, $\delta \in \{-2, -1, 0, +1, +2\}$, around each uniform reference $b \in \{2, 3, 4\}$ on $n{=}8192$ random Gaussian keys at $d{=}128$, averaged over 4 rotation seeds. Table 5 is the diagonal sweep with every entry expressed relative to the uniform (b, b) baseline. Because step δ spends $3b{+}\delta$ bits, App. G complements it with an exact-rate comparison on real keys.

E Rounding ablation

Table 6 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

Table 6: **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%



(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.

are averaged over five seeds on n=4096 random Gaussian keys at d=128 with n_{query}=64. *tail95* is 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 [24]: 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 Synthetic fidelity plots

G Real-key codec ablations

All experiments in this section run offline on rotated keys dumped from every layer and KV head of Qwen2.5-7B-Instruct-1M (28 layers, 4 KV heads), SmolLM3-3B (36, 4), and Mistral-7B-v0.3 (32, 8), all with d=128, on four 256-token prompts. Keys are split by (prompt, layer): calibrated baselines are fit on 4,096 calibration keys (65,536 triplets for the 3-D VQ), and all methods are evaluated on 1,024 held-out keys with 64 queries. MSE is the per-coordinate squared error of the reconstructed raw key (including its norm γ); because key norms differ strongly between models, absolute MSE is only comparable within a model.

G.1 Exact-rate comparison and bit allocation

Table 7 gives the absolute numbers behind Table 4 (top, left). The diagonal sweep of App. D compares budgets $3b{+}\delta$ against a $3b$ reference; this table instead fixes the deployed $B_{\text{tri}}{=}3b{+}1$ for both codecs. Table 8 lists every integer allocation of the octahedral codec at each budget. At $B_{\text{tri}}{=}3b{+}1$ the deployed $(b{+}1, b{-}1)$ split is the best allocation on all three families. At $B_{\text{tri}}{=}3b$ the feasible direction–norm gaps are only 0 and 3, and the uniform split wins, so the $\mathcal{O}(1)$ gap of Eq. 10 is realised at $3b{+}1$.

Table 7: **Exact-rate comparison.** Balanced per-coordinate Lloyd-Max vs. OCTOPUS $(b{+}1, b{-}1)$ at identical B_{tri}. Per-coordinate MSE on raw keys.

model	B_{tri}	scalar MSE	OCTOPUS MSE	change
Qwen2.5	7	10.8737	7.7748	-28.5%
Qwen2.5	10	3.4870	2.8816	-17.4%
Qwen2.5	13	0.8590	0.7484	-12.9%
SmolLM3	7	0.32139	0.30619	-4.7%
SmolLM3	10	0.09352	0.09022	-3.5%
SmolLM3	13	0.02611	0.02475	-5.2%
Mistral	7	0.31025	0.29341	-5.4%
Mistral	10	0.09047	0.08517	-5.9%
Mistral	13	0.02527	0.02323	-8.1%

Table 8: **Bit allocation within the octahedral class.** MSE of OCTOPUS (local 3×3 rounding) for every integer $(b_{\text{dir}}, b_{\text{nrm}})$ with $2b_{\text{dir}}{+}b_{\text{nrm}}{=}B_{\text{tri}}$; best per row in bold. Deployed splits are $(3, 1)$, $(4, 2)$, $(5, 3)$.

B_{tri}	Qwen2.5	SmolLM3	Mistral
6	**(2,2) 20.16**; (1,4) 65.44	**(2,2) 0.5277**; (1,4) 1.679	**(2,2) 0.5050**; (1,4) 1.609
7	**(3,1) 7.775**; (2,3) 18.79; (1,5) 65.31	**(3,1) 0.3062**; (2,3) 0.4887; (1,5) 1.675	**(3,1) 0.2934**; (2,3) 0.4682; (1,5) 1.605
9	(4,1) 4.754; **(3,3) 4.578**; (2,5) 18.40	(4,1) 0.2188; **(3,3) 0.1393**; (2,5) 0.4744	(4,1) 0.2083; **(3,3) 0.1346**; (2,5) 0.4546
10	**(4,2) 2.882**; (3,4) 4.208; (2,6) 18.37	**(4,2) 0.0902**; (3,4) 0.1273; (2,6) 0.4737	**(4,2) 0.0852**; (3,4) 0.1234; (2,6) 0.4539
12	(5,2) 1.897; **(4,4) 1.366**; (3,6) 4.078	(5,2) 0.0668; **(4,4) 0.0361**; (3,6) 0.1232	(5,2) 0.0627; **(4,4) 0.0346**; (3,6) 0.1197
13	(6,1) 3.544; **(5,3) 0.7484**; (4,5) 1.258	(6,1) 0.1906; **(5,3) 0.0248**; (4,5) 0.0328	(6,1) 0.1812; **(5,3) 0.0232**; (4,5) 0.0315

G.2 Calibrated 3-D vector quantization

The VQ baseline uses one global codebook per (model, rate), shared across all layers and heads, fit by generalized Lloyd (k-means) on 65,536 rotated triplets with 128 / 1,024 float32 centroids at $B_{\text{tri}}{=}7$ / 10, one deterministic initialization, and 20 iterations without restarts. Codebook storage is 1.5 / 12 KiB vs. 40 / 80 B for OCTOPUS.

G.3 Direction-encoding ablation

Spherical angles use a uniform azimuth plus a calibrated polar codebook; raw Cartesian uses balanced scalar Lloyd tables followed by renormalization; oct-pair and spherical VQ use $2^{2b_{\text{dir}}}$ calibrated 2-D and spherical centroids, respectively. Triplet decomposition, $(b{+}1, b{-}1)$ allocation, norm codebook, and total budget are held fixed. Table 10 reports all three families. The octahedral map beats spherical

Table 9: **Same-rate 3-D VQ vs. OCTOPUS.** MSE and inner-product mean absolute error (IP MAE) on held-out triplets.

model	B_{tri}	OCTOPUS MSE	VQ MSE	OCTOPUS IP MAE	VQ IP MAE
Qwen2.5	7	7.775	8.186	10.369	9.797
Qwen2.5	10	2.882	2.354	5.920	5.198
SmolLM3	7	0.3062	0.2454	4.973	4.463
SmolLM3	10	0.0902	0.0687	2.699	2.355
Mistral	7	0.2934	0.2357	4.865	4.385
Mistral	10	0.0852	0.0657	2.628	2.301

angles, raw Cartesian, and oct-pair VQ everywhere; the unstructured spherical VQ is within $\pm 1\%$ on SmolLM3 and Mistral. Fitting the spherical VQ to the uniform-sphere prior instead of calibration data gives $+8.3\%$ / -7.0% relative to OCTOPUS on Qwen2.5 (B_{tri}=7 / 10) and within $\pm 1\%$ elsewhere.

Table 10: **Direction-encoding ablation, all families.** MSE relative to OCTOPUS at the deployed split (calibrated alternatives); negative means the alternative is better.

model	B_{tri}	sph. angles	raw Cartesian	oct-pair VQ	sph. VQ
Qwen2.5	7	$+34.4\%$	$+46.6\%$	$+24.0\%$	$+4.9\%$
Qwen2.5	10	$+4.3\%$	$+85.1\%$	$+3.3\%$	$+2.1\%$
SmolLM3	7	$+14.2\%$	$+12.1\%$	$+9.6\%$	-0.7%
SmolLM3	10	$+18.9\%$	$+52.4\%$	$+5.5\%$	-0.3%
Mistral	7	$+17.3\%$	$+11.8\%$	$+6.2\%$	-0.0%
Mistral	10	$+12.9\%$	$+53.0\%$	$+6.4\%$	-0.2%

G.4 Head-dimension sensitivity and tail padding

Our real-key dumps all have d=128, so Table 11 sweeps the head dimension over a factor of eight on synthetic Gaussian keys at the deployed split. Reconstruction is essentially flat. Since $d \bmod 3 \neq 0$ for all four values, these numbers use the zero-padded tail block. On Qwen2.5 keys, coding the incomplete last block with a balanced scalar quantizer at the same budget instead of zero-padding it changes MSE by only 0.06% (B_{tri}=7) and 2.19% (B_{tri}=10). The d=512 point is reconstruction-only: the fused kernel is currently compiled for $d \leq 256$, a compile-time rather than algorithmic bound.

Table 11: **Head-dimension sweep.** Synthetic Gaussian keys, deployed $(b{+}1, b{-}1)$ split, per-coordinate MSE, and packed key size.

d	MSE B_{tri}=7	bytes/key	MSE B_{tri}=10	bytes/key
64	0.08584	24	0.02534	32
128	0.08327	42	0.02433	58
256	0.08543	80	0.02495	112
512	0.08407	154	0.02459	218

G.5 Combinability

OCTOPUS is a key-side codec rather than an indivisible cache design; Table 12 lists the combinations currently implemented. The octahedral direction encoding is orthogonal to value quantization, residual windows, token eviction, and boundary or layer-adaptive protection, so these stack on top of OCTOPUS without codec-level changes.

H QJL effective-rate accounting

Table 13 spells out the effective-rate cost of adding the one-bit residual side-car described in Sec. 3.6.

Table 12: **Combinability of OCTOPUS with other cache techniques.**

combination	status
OCTOPUS keys + independently quantized values	implemented (fused / group value decoders)
Full-precision residual window	implemented; merged with compressed history at decode
Optional 1-bit QJL residual (OCTOPUS-QJL)	implemented (Sec. 3.6, Algorithm 1)
Boundary-block protection	implemented (Sec. 4.2)
Per-layer / layer-adaptive precision	implemented via a per-layer cache policy
Alternative direction codebooks	reference only; needs new fused kernels
Full 3-D VQ direction encoding	reference / offline only; needs a new fused kernel

Table 13: **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

I Kernel speed and KV compression

Table 14 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 2.6–$4.6\times$ overhead vs. the highly optimised cuDNN SDPA bf16 path: 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. Relative to TurboQuant-MSE, the OCTOPUS kernel is 1.17–$1.45\times$ slower on video, 1.05–$1.07\times$ on audio, and 1.10–$1.37\times$ on the LLM. TQ-MSE encode is lightweight ($\leq$2 ms even at 65k tokens); OCTOPUS encode uses a Kronecker-factored WHT and direct triad indexing, bringing the per-token cost to $\approx 0.08\,\mu$s (d_h=128), which is negligible for auto-regressive LLM decode and small relative to diffusion denoising for video.

End-to-end decode. Table 15 reports end-to-end Qwen2.5-7B decode on a dedicated H200: 256 cached one-token forwards through all 28 layers with 32-token residual buffers, so cache updates, projections, layer overheads, and dispatch are included. At 64k context the KV footprint is 1.382 GiB for OCTOPUS, 1.344 GiB for TurboQuant-MSE, and 3.513 GiB for bf16 ($2.54\times$ reduction). OCTOPUS is 1.11–$1.16\times$ faster than TurboQuant-MSE at every context length, but both codecs reach only 0.24–$0.34\times$ of bf16 SDPA throughput. At batch size 1 this benchmark is dominated by per-step overheads of the Hugging Face integration (bf16 throughput is flat from 4k to 64k context, and the compressed codecs' decode time barely changes from 4k to 16k), not by KV bandwidth. The end-to-end ordering therefore reflects the integration path (cache update and dispatch) rather than the attention kernel, which is slower for OCTOPUS in isolation (Table 14).

Table 14: **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	0.07	1.0×	1.0×
4	OCTOPUS	3.8	0.29	4.4×	2.8×
4	TQ-MSE	2.0	0.20	3.0×	3.0×
3	OCTOPUS	3.6	0.24	3.6×	3.6×
3	TQ-MSE	2.1	0.21	3.1×	3.8×
2	OCTOPUS	3.6	0.23	3.5×	4.5×
2	TQ-MSE	1.9	0.18	2.6×	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	0.09	0.12	4.3×	2.4×
4	TQ-MSE	0.17	0.12	4.1×	2.6×
3	OCTOPUS	0.09	0.13	4.4×	2.9×
3	TQ-MSE	0.21	0.12	4.1×	3.0×
2	OCTOPUS	0.09	0.12	4.4×	3.5×
2	TQ-MSE	0.17	0.12	4.2×	3.8×
LLM (Qwen2.5-7B GQA, H_{kv}=4, d_h=128, g=32, T=65,536)					
–	SDPA bf16	0.11	0.07	1.0×	1.0×
4	OCTOPUS	5.2	0.33	4.6×	3.0×
4	TQ-MSE	1.5	0.24	3.4×	3.1×
3	OCTOPUS	4.6	0.27	3.7×	3.7×
3	TQ-MSE	1.6	0.24	3.4×	3.9×
2	OCTOPUS	4.5	0.25	3.5×	4.8×
2	TQ-MSE	1.5	0.20	2.7×	5.1×

Table 15: **End-to-end Qwen2.5-7B decode throughput.** Dedicated NVIDIA H200, batch size 1, b=3 (boundary-1 recipe), 256 cached one-token forwards, 28 layers, residual window 32, configurations interleaved round-robin. Tokens per second (↑, p50).

context	bf16 SDPA	TQ-MSE fused	OCTOPUS fused	OCTOPUS/ TQ-MSE
4k	64.82	19.65	21.95	1.12×
16k	64.38	19.66	21.89	1.11×
64k	64.58	15.44	17.84	1.16×

Isolated kernel overhead. In the isolated LLM kernel of Table 14 (H_{kv}=4, d=128, T=65,536), OCTOPUS is 1.10–1.37× slower than TurboQuant-MSE. Its key format is also slightly larger (3.34/4.31/5.41 vs. 3.13/4.13/5.13 effective bits per scalar at b=2/3/4; the integer formats do not expose an exact same-byte pair). At b=3 the octahedral fold adds 11.27M algorithmic operations, 1.50 MiB of key traffic, and 0.025 ms p50 per step.

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

Table 16 expands the long-context retrieval summary from Sec. 4.2 across the full context-length and bit-width grid.

K Downstream long-context evaluation

Table 17 reports the LongBench subset of Sec. 4.2 per task, together with WikiText-2 and C4 perplexity from the same bf16 evaluation harness. These perplexities use a shorter protocol than Table 2 (WikiText-2: 8 non-overlapping 512-token blocks; C4: 8 windows of 2,048 tokens with

Table 16: **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**

Table 17: **Downstream evaluation on `Qwen2.5-7B-Instruct-1M`.** 7-task LongBench subset (20 examples per task; official LongBench-v1 scoring, 0–100), WikiText-2 and C4 perplexity; symmetric $K{=}V$ at $b{=}3$.

codec	NQA	Qasper	MFQA	HQA	2WQA	GovRep	QMSum	**avg**	WT2 (↓)	C4 (↓)
bf16 baseline	15.24	23.59	21.95	12.20	9.48	16.24	19.22	16.85	9.851	13.753
TurboQuant-MSE	9.66	20.59	18.40	11.80	7.10	15.81	19.37	14.68	11.081	14.669
OCTOPUS	12.00	22.31	22.13	13.84	5.45	14.52	19.01	**15.61**	**11.024**	**14.480**

stride 1,024) and are therefore not directly comparable to it. All compressed runs use $b{=}3$, residual window 32, V group size 32, and the boundary-1 recipe.

L Memory-budget Pareto

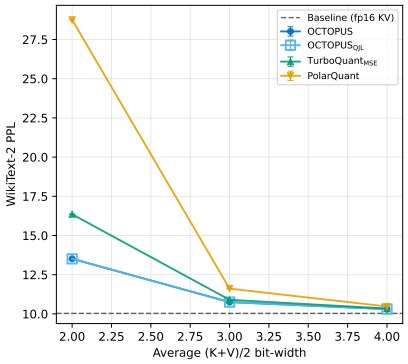
Figure 4 recasts the LLM results as a deployment memory trade-off, making the Pareto frontier visible at fixed context length.

M Full per-modality tables

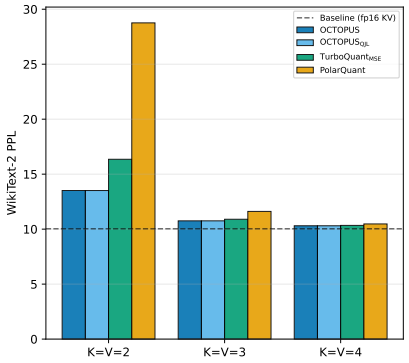
Table 18, Table 19, and Table 20 provide the per-codec metrics behind the modality summary in Sec. 4.3.

N Stills

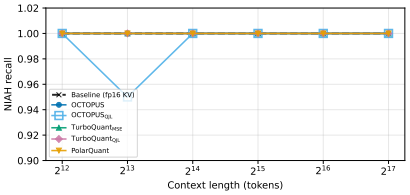
Figure 5 shows representative worst-case frames for the video codecs, complementing the aggregate LPIPS/PSNR/SSIM numbers in Sec. 4.3.



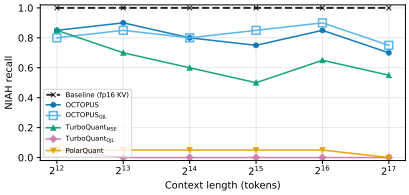
(a) PPL (↓) vs. bits



(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.

Table 18: **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**

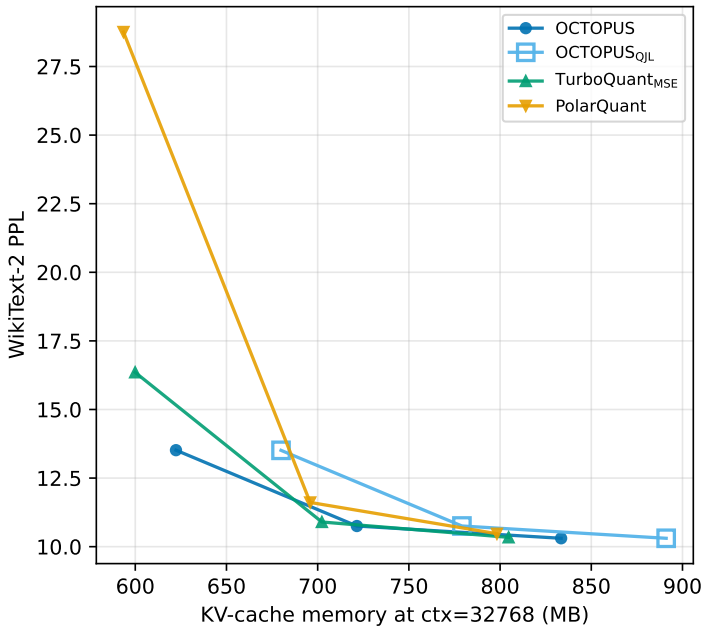


Figure 4: **LLM quality at fixed deployment memory.** WikiText-2 perplexity vs. KV-cache memory at 32,768-token context for `Qwen2.5-7B-Instruct-1M`. The probe-time `kv_cache_bytes` is linearly extrapolated from the sweep's measurement window to 32k tokens, so the x-axis is the memory budget a deployment 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 19: **Causal Forcing video generation, expanded.** Same recipe and conventions as Table 18.

bits	codec	compr.	LPIPS ↓	PSNR ↑	SSIM ↑	CLIP ↑	lat-cos ↑
4	TurboQuant-MSE	2.84×	0.3339	14.58	0.5547	0.3163	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×	0.3093	15.21	0.5757	0.3152	0.8192
4	**OCTOPUS-QJL**	2.48×	0.3103	15.23	0.5775	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×	0.3904	13.48	0.5090	0.3157	0.7594
3	**OCTOPUS-QJL**	2.96×	**0.3892**	**13.54**	**0.5123**	0.3158	**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×	0.7273	8.62	0.2155	0.2089	0.2984
2	**OCTOPUS**	4.05×	**0.5808**	**10.90**	**0.3593**	0.3024	**0.5618**
2	**OCTOPUS-QJL**	3.57×	**0.5808**	10.89	0.3577	**0.3034**	0.5592

Table 20: **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 underlined.

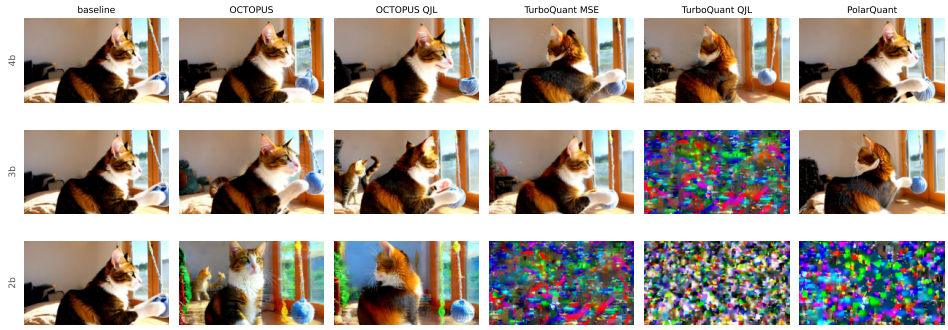
bits	codec	compr.	LSD ↓	mel-MSE ↓	SNR ↑	lat-cos ↑
4	TurboQuant-MSE	2.29×	6.36	0.2191	2.07	–
4	TurboQuant-QJL	2.26×	6.24	0.2284	1.80	–
4	PolarQuant	2.31×	6.28	0.2130	**2.23**	–
4	**OCTOPUS**	2.21×	6.20	0.2134	2.22	–
4	**OCTOPUS-QJL**	2.11×	**6.17**	**0.2104**	2.19	–
3	TurboQuant-MSE	2.48×	6.48	0.2377	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	0.2343	1.51	–
3	**OCTOPUS-QJL**	2.31×	6.43	0.2346	1.60	–
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×	6.88	0.3240	0.97	–



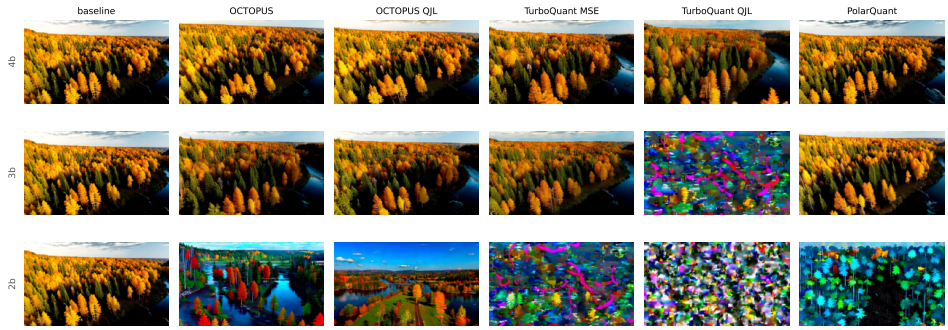
(a) CausVid



(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 $b{\leq}3$.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction claim an octahedral triplet codec, a non-uniform bit split, optional QJL residuals, fused decoding, and cross-modality gains; these are developed in Section 3, evaluated in Section 4, and supported by the ablations and derivations in the appendix.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 4.5 lists current limitations: the per-step compute overhead of the fused decode kernel relative to scalar dequantization and bf16 SDPA (quantified in Section I), the lower MSE attainable by a per-model calibrated 3-D VQ (Section 4.4), the fixed block size of three with tail padding, and the current $d \leq 256$ bound of the fused kernel.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The main text states the rotation, sphere-prior, triplet, Lloyd-Max, and ideal-QJL assumptions, while Section B and the derivation appendix give the proofs and derivations for the marginal, MSE, bit-allocation, score-factorization, and QJL-estimator claims.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 4 specifies the compared codecs, model families, bit widths, context lengths, residual windows, value group sizes, metrics, prompt counts, and seed averaging; Section A gives reference encoder and decoder algorithms sufficient to reproduce the mathematical codec behavior.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Code, including the reference implementation and fused Triton kernels, will be available via `https://octopus-quant.github.io/`. All evaluation data are existing public datasets (Section 4); Section A gives the encoder and decoder algorithms.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The experiments are inference-only and Section 4 gives the relevant evaluation settings: models, dimensions, context lengths, bit widths, residual windows, value group sizes, boundary protection, prompt counts, metrics, and the synthetic and needle-in-a-haystack protocols.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

Justification: The paper reports averages over seeds and shared prompt sets for several experiments.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section I reports kernel timing on a single NVIDIA H200.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification:

   Guidelines:

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

10. **Broader impacts**

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

Answer: [No]

Justification: Section 5 notes that OCTOPUS enables slightly more efficient, slimmed-down deployment of existing autoregressive models and may draw more attention to efficiency research; it creates no new model capabilities nor does it allow significantly easier access to negative or unintended usage of existing models.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification:

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All existing assets are cited in Section 4 and used solely for research evaluation, without redistribution, in accordance with their terms: Qwen2.5-7B-Instruct-1M, SmolLM3-3B, Mistral-7B, Wan2.1-T2V-1.3B, and the CLAP encoder `laion/larger_clap_general` (Apache-2.0); Causal Forcing (code Apache-2.0, checkpoints MIT); CausVid (code CC BY-NC-SA 4.0, weights CC BY-NC 4.0, used non-commercially); AAR (public research code and checkpoints, released without an explicit license, built on AudioSet); AudioSet (annotations CC BY 4.0, audio subject to YouTube terms of service); WikiText-2 (CC BY-SA 3.0); C4 (ODC-BY); and LongBench (MIT).

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The only new asset is our code (reference implementation, fused Triton kernels, and evaluation scripts), which will be available with documentation via `https://octopus-quant.github.io/`. We introduce no new datasets.

Guidelines:

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

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

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

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification:

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: No LLM is a component of the proposed method. LLMs (Qwen2.5-7B-Instruct-1M, SmolLM3, Mistral) do appear as major evaluation targets whose KV caches are compressed, as fully described in Section 4.

    Guidelines:

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