METRICKV: DECODED-SPACE GEOMETRY FOR VISUAL KV COMPRESSION IN STREAMING VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video models observe visual evidence before future queries are known, making compact storage of historical visual key–value (KV) states important for long-running inference. We introduce METRICKV, a query-agnostic visual-KV codec derived from the decoded-KV reconstruction objective. For any positive-definite quadratic decoded-space distortion and full-column-rank decoder, decoded error decomposes exactly into a continuous projection term and an additional finite-rate latent error under a decoder-induced pullback metric. This metric determines the same ordering over finite-rate candidates as the decoded-KV objective; absorbing it into the latent coordinates yields a canonical Euclidean coding space in which standard nearest-neighbor quantization is exactly decoded-error aligned. Within this canonical family, minimizing the coder-independent continuous projection term reduces to PCA in the distortion-transformed KV space. We instantiate the framework with Layer-RMS reconstruction geometry and progressive residual vector quantization (RVQ). The experimental results demonstrate that a single progressive RVQ stream achieves $64\times$ dynamic-KV compression while retaining approximately 98% of FP16 performance on both OVO-Bench and Streaming-Bench. Code will be released soon.

1 INTRODUCTION

Multimodal large language models (MLLMs) increasingly process video as a continuous stream, where future queries may depend on visual evidence observed much earlier (Di et al., 2025; Zhang et al., 2026; Qian et al., 2024; Yang et al., 2025; Chen et al., 2026; Shen et al., 2026). Visual key–value (KV) states provide a natural persistent memory because they can be reused without re-encoding historical frames, but their storage grows continuously with stream duration. Existing streaming systems primarily address what visual evidence should be retained, summarized, or retrieved. We study a complementary problem: given visual KV that a streaming system chooses to preserve, how should it be represented compactly for later reuse?

A common compression pipeline first projects KV into a low-dimensional latent and then quantizes that latent for storage, as illustrated in Figure 1 (a). Once the continuous latent $z = Ex$ is obtained, a finite-rate coder must choose among a discrete set of reconstructions. This choice is typically made using a distance defined directly in latent space, although reconstruction quality is ultimately determined after decoding back to KV space. These two criteria need not rank the same candidates consistently: a reconstruction $\hat{z}_A$ can be closer to z than $\hat{z}_B$ under the chosen latent metric, yet incur larger decoded-KV error after applying the decoder, as illustrated in Figure 1 (b). The mismatch therefore changes not only the measured error but potentially the discrete reconstruction selected and stored by the codec. This raises two related questions: *what latent geometry makes finite-rate decisions consistent with decoded-KV reconstruction error, and, once this geometry is fixed, which rank-r subspace should be retained?*

We derive both from the decoded-KV objective. For a fixed quadratic decoded-space distortion and decoder, we show that reconstruction error decomposes exactly into a continuous rank-r projection term and an additional finite-rate latent error measured by a decoder-induced pullback metric. Consequently, this metric ranks every finite-rate candidate exactly as the decoded-KV objective does. We

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

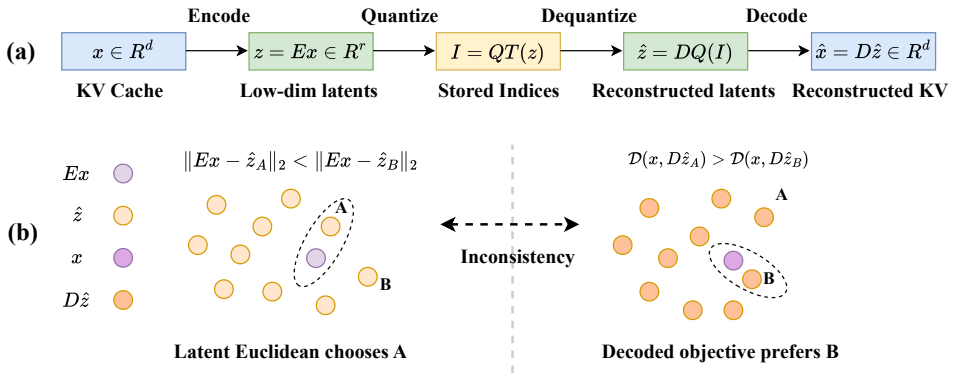


Figure 1: Motivation of decoded-space geometry for visual-KV compression. (a) Visual KV is projected to a low-dimensional latent, discretized into stored indices, and reconstructed when historical evidence is reused. (b) A latent distance chosen independently of the decoder need not rank finite-rate reconstructions consistently with their actual decoded-KV error: a candidate that is closer in latent space can incur larger error after decoding. This mismatch motivates deriving the finite-rate latent geometry directly from the decoded-space reconstruction objective.

absorb the metric into the latent coordinates, producing a canonical Euclidean coding space in which ordinary nearest-neighbor quantization assignments are decoded-error aligned. Canonicalization fixes the coding geometry; across the resulting canonical family, the retained rank-r subspace remains to be chosen. Because the finite-rate term additionally depends on the coder, rate, and fitted codebooks, jointly optimizing the subspace for finite-rate performance is coder- and rate-dependent. We therefore minimize the coder-independent continuous projection term, which reduces to the classical PCA (Jolliffe, 2002) solution in the distortion-transformed KV space.

To instantiate the framework for streaming visual KV, the decoded-space objective must be specified before future queries are known. We use a query-agnostic Layer-RMS objective that measures reconstruction error relative to the typical activation scale of each layer, and use progressive RVQ (Juang & Gray, 1982) as the finite-rate coder. Codec statistics, linear transforms, and codebooks are constructed offline from unlabeled visual KV from LongVideoBench (Wu et al., 2024). During streaming inference, retained KV is encoded once into the canonical latent and stored as a sequence of RVQ indices. When the existing memory policy later requests historical evidence, the corresponding indices are dequantized and decoded back to KV for reuse by the unchanged MLLM; different storage budgets are obtained by truncating the same progressive index stream. Empirically, the proposed METRICKV achieves strong compression–accuracy trade-offs across streaming video benchmarks OVO-Bench (Niu et al., 2025) and StreamingBench (Lin et al., 2026) with a single progressive RVQ stream. Applying the same Layer-RMS reconstruction objective to KVTC (Staniszewski & Lancucki, 2026), without changing its scalar coder, also improves performance across tested rates, supporting the separation between decoded-space reconstruction geometry and finite-rate coding. METRICKV is orthogonal to the surrounding streaming memory policy and can be combined with different retention or retrieval strategies.

Our contributions are:

- We formulate finite-rate visual-KV compression from a decoded-space quadratic objective and derive the decoder-induced pullback metric, yielding an exact decomposition into continuous projection error and finite-rate coding error with consistent candidate ordering across latent and decoded spaces.
- We canonicalize this metric into a Euclidean latent space and characterize the remaining rank-r subspace choice, whose coder-independent continuous optimum reduces to PCA in the distortion-transformed KV space.
- We instantiate the framework as METRICKV with progressive RVQ for streaming visual KV, achieving strong compression–accuracy trade-offs across compression rates and benchmarks while substantially reducing KV-cache storage.

2 Related Work

Streaming video memory. Streaming video understanding (Qian et al., 2025; Zhang et al., 2026; Zeng et al., 2026; Chen et al., 2025) requires controlling a visual memory that grows with the input stream. Existing systems primarily address this by deciding what visual evidence to retain or retrieve: ReKV (Di et al., 2025) stores visual KV blocks and retrieves query-relevant ones at answer time; StreamMem (Yang et al., 2025) maintains a bounded query-agnostic memory through attention-guided token compression; StreamKV (Chen et al., 2026) and MuKV (Xiao et al., 2026) organize memory through semantic or hierarchical structures for selective retrieval. SimpleStream (Shen et al., 2026) further shows that a recent-window policy provides a strong baseline. These approaches operate mainly at the memory-management level, whereas MetricKV compresses the numerical representation of retained visual KV. The two are complementary: MetricKV can operate beneath different retention or retrieval policies to reduce the storage cost of each retained state.

KV-cache compression. Prior work reduces KV-cache cost through low-bit quantization, low-rank projection and token removal. KIVI (Liu et al., 2024), KVQuant (Hooper et al., 2024), GEAR (Kang et al., 2024), PALU (Chang et al., 2025) and Coupled Quantization (Zhang et al., 2024) improve finite-precision KV representations via asymmetric quantization, error correction, or cross-dimensional transforms. KVTC (Staniszewski & Lancucki, 2026) combines PCA-based transform coding with adaptive scalar quantization and entropy coding, while VL-Cache and FlashCache (Tu et al., 2025; Yang et al., 2026) reduce cache length by discarding selected positions. MetricKV starts from a prescribed decoded-KV reconstruction objective and derives the corresponding finite-rate geometry in latent space. This formulation separates the decoded-space objective, retained continuous subspace, and finite-rate coder.

Discrete representation learning. Vector quantization maps continuous features to finite codebooks (Van Den Oord et al., 2017), while residual vector quantization (RVQ) successively quantizes residuals with multiple codebooks (Juang & Gray, 1982), enabling progressive or variable-rate reconstruction (Zeghidour et al., 2021; Défossez et al., 2022). MetricKV uses RVQ only as one finite-rate coding mechanism. Unlike learned VQ models that jointly optimize the representation and codebooks (Van Den Oord et al., 2017; Zeghidour et al., 2021; Défossez et al., 2022), our continuous representation geometry is fixed analytically before codebook refinement. The same geometry can therefore be paired with a different finite-rate coder, which we test by transferring Layer-RMS scaling to KVTC without changing its scalar quantization mechanism.

3 Method

3.1 Overview and Compression Formulation

MetricKV compresses visual key–value (KV) states from a frozen MLLM without modifying the backbone or attention computation. For each visual token, we jointly represent KV states from a fixed group $b = \{\ell_1, \dots, \ell_P\}$ of P transformer layers and a group $g = \{h_1, \dots, h_G\}$ of G KV heads:

$$x = \mathrm{Concat}_{\ell \in b}\, \mathrm{Concat}_{h \in g}\, [K_{\ell,h}, V_{\ell,h}] \in \mathbb{R}^d, \qquad d = 2PGd_h, \tag{1}$$

where d_h is the per-head key/value dimension. A separate codec is constructed for each layer-group–head-group bucket (b, g), while all tokens assigned to the same bucket share its codec parameters.

We consider a rank-r finite-rate codec of the form:

$$z = Ex, \qquad \hat{z} = Q(z), \qquad \hat{x} = D\hat{z}, \tag{2}$$

where $E \in \mathbb{R}^{r \times d}$ and $D \in \mathbb{R}^{d \times r}$ are the encoder and decoder, Q is a finite-rate coder, and $r < d$. Before compression, we subtract the empirical mean of visual KV features. For simplicity of notation, we use x to denote the centered feature throughout the paper. The corresponding mean is stored as a fixed codec parameter and added back after decoding.

The design problem is therefore not only which rank-r representation to retain, but also how finite-rate error in the latent space should be measured so that discrete coding decisions remain consistent with the reconstruction objective in decoded KV space.

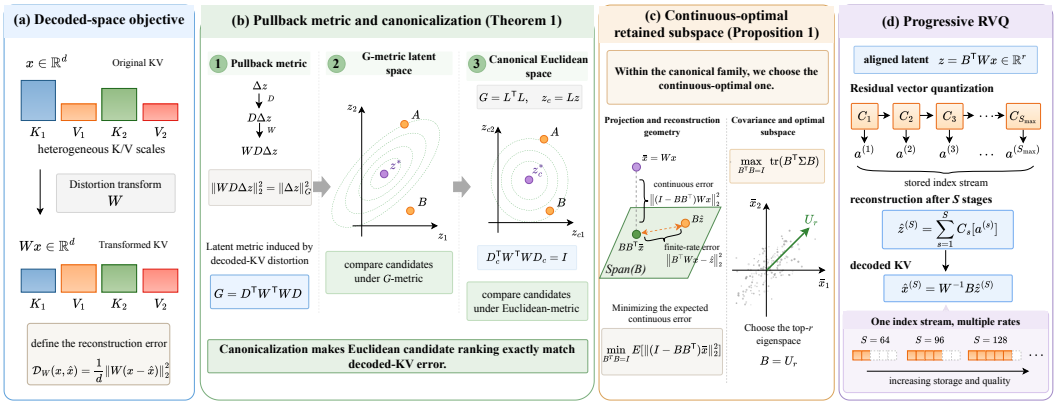


Figure 2: **Overview of METRICKV.** (a) A decoded-space transform (W) defines the reconstruction objective; Layer-RMS is our default instantiation. (b) The decoder induces the latent pullback metric G, which is canonicalized into Euclidean coordinates so that finite-rate decisions match decoded-KV error. (c) Within this aligned family, the top-r eigenspace of the transformed KV distribution minimizes coder-independent continuous projection error. (d) Progressive RVQ discretizes the resulting latent into a single index stream supporting multiple storage–quality operating points.

We first specify the decoded-space distortion, then derive and canonicalize its decoder-induced latent metric, and finally optimize the remaining continuous rank-r subspace. This yields PCA in the distortion-transformed KV space, after which any compatible finite-rate coder can be applied. We instantiate this final stage with progressive RVQ, which represents each latent using a single ordered sequence of residual codebook indices and supports multiple storage–quality operating points by truncating the retained prefix.

3.2 DECODED-SPACE DISTORTION

Our objective is to preserve the reconstructed visual KV rather than the latent representation itself. We therefore define distortion directly in decoded KV space. Let $W \in \mathbb{R}^{d\times d}$ be a fixed invertible transform and define:

$$\mathcal{D}_W(x,\hat{x}) = \frac{1}{d}\,\|W(x-\hat{x})\|_2^2\,. \tag{3}$$

Here, W specifies how reconstruction errors in different KV components are weighted. Raw MSE is recovered by $W = I_d$. All following derivations hold for any fixed invertible W; thus, W defines the decoded-space reconstruction objective independently of the subspace and finite-rate coder.

For visual KV, we instantiate this objective with blockwise scale normalization. Let $x = \mathrm{Concat}_{j=1}^{J} x_j$, where $x_j \in \mathbb{R}^{d_j}$ denotes one KV block, and define its population RMS as $s_j = \sqrt{\frac{1}{d_j}\mathbb{E}\left[\|x_j\|_2^2\right]}$. Using

$$W = \mathrm{diag}\left(\frac{1}{s_1}I_{d_1},\ldots,\frac{1}{s_J}I_{d_J}\right), \qquad \mathcal{D}_W(x,\hat{x}) = \frac{1}{d}\sum_{j=1}^{J}\frac{\|x_j-\hat{x}_j\|_2^2}{s_j^2}, \tag{4}$$

measures reconstruction error relative to the typical magnitude of each block. This prevents blocks with larger activation scales from dominating the objective solely because of their numerical magnitude and makes the distortion invariant to independent rescaling of the blocks.

Our default, *Layer-RMS*, assigns one scale to each transformer layer, making the reconstruction objective invariant to layer-wise scale differences while preserving the relative weighting among heads and between keys and values within each layer. Details can be found in Appendix B.

3.3 FROM DECODED-KV ERROR TO A CANONICAL LATENT SPACE

Once the decoded-space distortion $\mathcal{D}_W$ is fixed, a rank-r codec still involves two distinct design choices: *how finite-rate latent reconstructions should be compared*, and *which rank-r continuous*

subspace should be retained. To address these two choices, we first derive the latent metric exactly induced by decoded-KV distortion for an arbitrary decoder, then reparameterize the latent space so that this metric becomes Euclidean, and finally select the retained subspace over the resulting family of canonical encoder–decoder pairs using the coder-independent continuous reconstruction error.

1. Pulling decoded-KV error into latent space. Fix a full-column-rank decoder $D \in \mathbb{R}^{d\times r}$. Before quantization, the best latent representation of x under the decoded-space objective is the weighted least-squares solution:

$$z^\star = \operatorname*{argmin}_{z\in\mathbb{R}^r} \mathcal{D}_W(x, Dz) = Ex, \qquad E = G^{-1}D^\top W^\top W, \qquad G = D^\top W^\top W D \succ 0. \tag{5}$$

A finite-rate coder replaces this continuous optimum $z^\star$ by a discrete reconstruction $\hat{z}$. The following result characterizes exactly how the resulting coding error is measured in latent space.

Theorem 1 (Decoded-distortion pullback) *For any source vector x and latent reconstruction $\hat{z}$,*

$$\mathcal{D}_W(x, D\hat{z}) = \mathcal{D}_W(x, Dz^\star) + \frac{1}{d}\,\|z^\star - \hat{z}\|_G^2\,, \qquad z^\star = Ex. \tag{6}$$

Consequently, for any candidate set $\mathcal{C} \subseteq \mathbb{R}^r$,

$$\operatorname*{argmin}_{\hat{z}\in\mathcal{C}} \|z^\star - \hat{z}\|_G^2 = \operatorname*{argmin}_{\hat{z}\in\mathcal{C}} \mathcal{D}_W(x, D\hat{z}). \tag{7}$$

Appendix A.1 gives the proof. Equation 6 separates the decoded distortion into two terms. The first is the minimum continuous reconstruction error attainable with the fixed decoder D and is independent of the finite-rate coder. The second is exactly the additional distortion introduced by replacing $z^\star$ with $\hat{z}$. Therefore, finite-rate latent reconstructions should be compared using $G = D^\top W^\top W D$ rather than an independently chosen latent distance. Indeed, any latent perturbation Δz induces $\|WD\Delta z\|_2^2 = \|\Delta z\|_G^2$ Thus, G is the decoded-space distortion pulled back through the decoder.

2. Canonical Euclidean coordinates. Theorem 1 gives the exact latent metric $G = D^\top W^\top W D$ induced by decoded-KV distortion, whereas standard vector quantization uses Euclidean distance. We therefore absorb G into the latent coordinates. Since $G \succ 0$, let $G = L^\top L$ and define

$$z_c = Lz, \qquad E_c = LE, \qquad D_c = DL^{-1}. \tag{8}$$

This change of coordinates leaves the decoded reconstruction unchanged, $D_c E_c = DE$, and yields

$$D_c^\top W^\top W D_c = I_r, \qquad E_c = D_c^\top W^\top W. \tag{9}$$

Hence Eq. 6 becomes

$$\mathcal{D}_W(x, D_c\hat{z}_c) = \mathcal{D}_W(x, D_c E_c x) + \frac{1}{d}\|E_c x - \hat{z}_c\|_2^2. \tag{10}$$

Thus, in the canonical coordinates, ordinary Euclidean nearest-neighbor assignment exactly preserves the candidate ordering induced by decoded-KV distortion.

3. Selecting a continuous-optimal retained subspace. The construction above canonicalizes any fixed decoder without changing its continuous decoded reconstruction. We now optimize over the full family of encoder–decoder pairs satisfying Eq. 9. Dropping the subscript c, define $\bar{x} = Wx$ and $B = WD$. These conditions give the equivalent parameterization

$$D = W^{-1}B, \qquad E = B^\top W, \qquad B^\top B = I_r. \tag{11}$$

Conversely, any B with $B^\top B = I_r$ defines such a canonical pair, so the remaining continuous design variable is $\mathrm{span}(B)$. Under this parameterization, Eq. 10 becomes

$$\mathcal{D}_W(x, D\hat{z}) = \frac{1}{d}\|(I - BB^\top)\bar{x}\|_2^2 + \frac{1}{d}\|B^\top\bar{x} - \hat{z}\|_2^2. \tag{12}$$

The first term is the continuous rank-r approximation error, while the second is the additional finite-rate error. Because the latter also depends on the coding mechanism, rate, and fitted codebooks, we select the subspace using the coder-independent continuous term.

Let $\Sigma = \mathbb{E}[\bar{x}\bar{x}^\top]$ be the covariance matrix of the transformed source. The continuous subspace objective can then be written equivalently as

$$\min_{B^\top B = I_r} \mathbb{E}\left[\frac{1}{d}\left\|(I - BB^\top)\bar{x}\right\|_2^2\right] \quad\Longleftrightarrow\quad \max_{B^\top B = I_r} \mathrm{tr}(B^\top \Sigma B). \tag{13}$$

Proposition 1 (Continuous-optimal canonical subspace) *Let $U_r \in \mathbb{R}^{d\times r}$ contain orthonormal eigenvectors associated with the largest r eigenvalues of Σ. Then $B = U_r$ minimizes the expected continuous rank-r reconstruction error over the canonical family.*

This is the classical PCA solution and follows from the Ky Fan maximum principle (Fan, 1949); see Appendix A.3. Substituting $B = U_r$ gives the final encoder and decoder

$$E = U_r^\top W, \qquad D = W^{-1}U_r. \tag{14}$$

Together with Theorem 1, this separates the two roles of the representation design: the theorem determines the finite-rate latent geometry induced by decoded-space distortion, while the proposition selects the continuous-optimal retained subspace under the same objective.

3.4 Progressive Residual Vector Quantization

The construction above is independent of the finite-rate coder. We use progressive residual vector quantization (RVQ) as our main instantiation because a single ordered index stream supports multiple storage–quality operating points: different rates are obtained by retaining different prefixes without changing the encoder, decoder, or codebooks.

For each bucket (b, g), we learn $S_{\max}$ codebooks $C_{b,g,s} \in \mathbb{R}^{M\times r}$, where M is the size of codebook. Starting from $r^{(0)} = z$, stage s selects

$$a^{(s)} = \underset{m\in\{1,\dots,M\}}{\operatorname{argmin}} \left\| r^{(s-1)} - C_{b,g,s}[m] \right\|_2^2, \qquad r^{(s)} = r^{(s-1)} - C_{b,g,s}[a^{(s)}]. \tag{15}$$

Using the first S stages gives

$$Q_S(z) = \hat{z}^{(S)} = \sum_{s=1}^{S} C_{b,g,s}[a^{(s)}], \qquad \hat{x}^{(S)} = D\hat{z}^{(S)}. \tag{16}$$

Because Eq. 14 satisfies $D^\top W^\top W D = I_r$, Euclidean distance in the resulting latent coordinates is exactly the pullback of the decoded-space distortion. At each RVQ stage, Euclidean nearest-neighbor selection therefore ranks candidate residual reconstructions consistently with their decoded-KV distortion; Appendix C gives more details about RVQ.

Each stage stores one $\log_2 M$-bit index for a joint vector spanning P layers and G KV heads. An S-stage prefix therefore requires

$$B_{\mathrm{dyn}}(S, P, G) = \frac{S \log_2 M}{8PG} \quad \text{bytes/head/layer/token.} \tag{17}$$

This is the dynamic storage of the packed RVQ indices. The encoder, decoder, normalization statistics, and codebooks are static codec parameters shared across visual tokens and stream duration; their storage is reported separately in Section 4.

Offline codec construction. As in Algorithm 1, we first estimate the Layer-RMS statistics from unlabeled visual KV and construct W. Then we transform the samples accordingly, and compute the top-r eigenspace of their covariance matrix. The encoder and decoder are constructed analytically using Eq. 14. These transforms are fixed thereafter. In the resulting canonical latent space, the RVQ codebooks are initialized sequentially by residual k-means and subsequently refined across the supported prefix lengths, while keeping E and D frozen. Codec construction uses only visual-KV samples, without downstream labels, future queries, or gradients through the frozen MLLM.

Streaming inference. When a visual token is written to memory, its grouped KV vector x is encoded once as $z = Ex$ and quantized sequentially by RVQ; only the resulting codebook indices are stored as the dynamic state. When the corresponding historical evidence is later requested, the retained prefix of indices is dequantized to $\hat{z}^{(S)}$ and decoded as $\hat{x}^{(S)} = D\hat{z}^{(S)}$ before being restored to the original KV layout and consumed by the unchanged MLLM. Different storage budgets use different prefixes of the same index stream, without changing the encoder, decoder, or codebooks.

Algorithm 1 Offline codec construction

Require: Unlabeled grouped KV samples $\{x_n\}$, rank r, codebook size M, prefixes $\mathcal{S}$
1: Estimate Layer-RMS statistics and construct W
2: Compute $\bar{x}_n = Wx_n$ and $\Sigma = \frac{1}{N}\sum_n \bar{x}_n \bar{x}_n^\top$
3: Compute the leading r eigenvectors U_r of Σ
4: Set $E = U_r^\top W$ and $D = W^{-1}U_r$; freeze E and D
5: Encode fitting samples as $z_n = Ex_n$
6: Initialize $S_{\max}$ RVQ codebooks sequentially by residual k-means
7: Refine only the codebooks
8: **return** W, E, D, and the ordered RVQ codebooks

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Evaluation protocol. We use the frozen Qwen3-VL-4B-Instruct (Bai et al., 2025) backbone and process videos causally at 1 fps. (More results on other backbones can be found in Appendix D.) We evaluate METRICKV at two levels. For the controlled compression comparison in Table 1, visual KV is generated in five-frame chunks and each query uses the same four externally retrieved historical frames and four recent frames across all methods. Thus, the selected evidence, ordering, positional treatment, and answer-time visual-token budget are fixed, and only the visual-KV representation varies. Table 2 instead evaluates complete streaming systems under their native memory policies.

Codec fitting and baselines. We fit METRICKV only on 500k unlabeled visual-KV tokens sampled from LongVideoBench (Wu et al., 2024), without using questions or answers, and directly evaluate the frozen codec on OVO-Bench (Niu et al., 2025) and StreamingBench (Lin et al., 2026). Unless otherwise stated, we use head-group size $G = 4$, layer-group size $P = 6$, latent rank $r = 3072$, and codebook size $M = 16$. We compare against FP16, scalar quantization, representative KV compression methods, and representative streaming-video systems.

Metrics. We report backward, real-time, and average performance on OVO-Bench and the official overall score on the real-time subset of StreamingBench following previous methods (Zhang et al., 2026; Shen et al., 2026). We exclude task HLD in OVO-Bench because all HLD targets are "Unable to answer," causing degraded visual memory to spuriously improve refusal accuracy. KV storage is reported as dynamic bytes per original head, layer, and visual token, including method-specific dynamic side information, with compression ratios relative to FP16. Controlled reconstruction experiments use the selected decoded-space distortion, with Layer-RMS as the default unless stated otherwise.

4.2 STREAMING VIDEO UNDERSTANDING

We evaluate METRICKV at two levels. We first isolate KV representation quality under a fixed visual readout, then evaluate the same codec in streaming systems using different memory policies.

Controlled KV compression under fixed readout. To isolate the effect of KV representation, all methods receive the same visual evidence: four retrieved historical frames and four recent frames, with identical ordering and answer-time visual-token budgets. Table 1 therefore compares only how the retained visual KV is represented.

At 8 B per head/layer/token ($64\times$ compression), METRICKV achieves 69.04 on OVO-Bench and 79.10 on StreamingBench, within 0.90 and 1.64 points of FP16, respectively. The 4-, 8-, and 16-B operating points use the same encoder, decoder, and progressive RVQ stream; only the retained RVQ prefix changes.

Integration with streaming memory policies. Table 2 evaluates METRICKV under two different memory policies while keeping evidence selection unchanged. For ReKV+METRICKV, we reuse the

Table 1: Controlled comparison of KV representations under an identical visual readout of four retrieved historical frames and four recent frames. Storage denotes dynamic bytes per original KV head, layer, and visual token; compression is relative to FP16.

Method	Storage	Compression	OVO-Bench			StreamingBench
			Real-time	Backward w/o HLD	OVO Avg.	
FP16	512	1.00×	81.03	58.85	69.94	80.74
INT8	272	1.88×	79.92	58.17	69.05	80.90
INT4	144	3.56×	80.41	58.00	69.21	80.74
INT2	96	5.33×	65.72	47.37	56.55	65.77
KIVI (Liu et al., 2024)	99	5.17×	80.20	57.84	69.02	80.66
PALU INT2 (Chang et al., 2025)	33	15.52×	72.80	51.27	62.03	73.02
PALU INT3 (Chang et al., 2025)	49	10.45×	79.77	58.17	68.97	80.70
CommVQ 0.5-bit (Li et al., 2025)	24	21.33×	68.41	44.86	56.64	68.33
CommVQ 1-bit (Li et al., 2025)	33	15.52×	78.89	56.66	67.77	78.30
CommVQ 2-bit (Li et al., 2025)	64	8.00×	79.50	57.84	68.67	80.14
KVQuant 1-bit (Hooper et al., 2024)	32	16.00×	51.35	42.00	46.68	52.84
KVQuant 2-bit (Hooper et al., 2024)	64	8.00×	78.85	55.31	67.08	77.98
KVTC (Staniszewski & Lancucki, 2026)	4	128.00×	78.88	55.48	67.18	77.94
KVTC (Staniszewski & Lancucki, 2026)	8	64.00×	79.77	56.32	68.04	**79.90**
KVTC (Staniszewski & Lancucki, 2026)	16	32.00×	79.87	57.33	68.60	**80.26**
METRICKV (S192)	4	128.00×	78.88	57.16	**68.02**	**78.70**
METRICKV (S384)	8	64.00×	79.57	58.51	**69.04**	79.10
METRICKV (S768)	16	32.00×	79.32	58.17	**68.75**	79.78

Table 2: System-level comparison on streaming video understanding. *Persistent State* reports the video-dependent state maintained across the stream: the native capacity for bounded-memory methods, and the storage growth rate for methods that retain the full history. Growth rates are measured at 1 fps with 576 visual tokens per frame and exclude model parameters and static codec parameters.

System	Persistent State	OVO-Bench			StreamingBench Avg.
		Real-time	Backward	Avg.	
Dispider-7B (Qian et al., 2025)	Growing (pooled clips)	54.55	36.06	45.31	67.63
TimeChat-Online-7B (Yao et al., 2025)	6K tokens (fixed)	61.90	41.70	51.80	75.36
StreamForest-7B (Zeng et al., 2026)	8K tokens (fixed)	61.20	52.02	56.61	77.26
HERMES-4B (Zhang et al., 2026)	4K tokens (fixed)	72.32	55.03	63.68	78.24
ReKV (Di et al., 2025)	4.746 GiB/min	64.74	50.59	57.66	69.66
ReKV + MetricKV(S384)	0.0742 GiB/min	63.59	50.93	57.26	67.01
SimpleStream-4B (Shen et al., 2026)	4 frames (fixed)	81.82	54.13	67.98	80.90
SimpleStream-4B + MetricKV(S384)	4 frames (fixed)	81.50	53.62	67.56	79.86

historical chunks selected by the ReKV reference; for SimpleStream+METRICKV, we preserve the original four-frame recent window. Thus only the representation of retained KV has been changed.

With the same frozen S384 codec (8 B per head/layer/token; $64\times$ compression), METRICKV reduces ReKV's persistent-state growth from 4.746 to 0.0742 GiB/min, while the SimpleStream budget remains fixed at four frames. Performance remains close to the corresponding FP16 systems, showing that METRICKV can operate beneath both retrieval-based and bounded-window memory policies.

Codec runtime and static memory. Table 4 reports the end-to-end cost of the packed-index codec. Writing includes KV grouping, encoder projection, greedy RVQ assignment, and 4-bit index packing; reading includes index unpacking, codeword accumulation, decoder projection, and KV-layout restoration. Measurements use one 576-token visual frame on a single RTX 3090 and report our optimized codec-only implementation.

Runtime increases approximately with the retained RVQ prefix length. Greedy RVQ assignment accounts for 93.9–98.4% of the measured write components, which is the primary write-side bottleneck. After fusing 4-bit unpacking, codeword lookup, and FP32 accumulation, the fused stage accounts for 22.6–60.2% of the read components. The reported 1.898–2.531 GiB static codec storage consists of the shared encoder/decoder transforms, RVQ codebooks, together with other fixed codec metadata. These parameters are independent of stream length and are therefore separate from the video-dependent persistent state reported in Table 2.

Table 3: Effect of the decoded-space objective across finite-rate coders on StreamingBench. We keep each finite-rate coding mechanism fixed and change only the decoded-space objective.

Coder	Decoded-space objective	4B	8B	16B
KVTC	Raw MSE ($W = I$)	77.94	79.90	80.26
	Layer-RMS ($W = W_{\mathrm{RMS}}$)	**79.26**	**80.10**	**80.58**
RVQ	Raw MSE ($W = I$)	77.58	78.86	79.70
	Layer-RMS ($W = W_{\mathrm{RMS}}$)	**78.70**	**79.10**	**79.78**

Table 4: Runtime and static codec size of METRICKV. Write and read latency are measured per 576-token visual frame on one RTX 3090.

Storage	Write/frame	Read/frame	Static Size
4 B	0.240 s	0.020 s	1.898 GiB
8 B	0.467 s	0.027 s	2.109 GiB
16 B	0.923 s	0.040 s	2.531 GiB

4.3 ABLATION STUDY

Effect of the decoded-space objective. Table 3 isolates the effect of the decoded-space objective from the finite-rate coding mechanism. For each coder, we keep the compression pipeline fixed and replace only Raw MSE with Layer-RMS. Layer-RMS improves KVTC at all three storage budgets, with the largest gain at 4B, and also improves RVQ at 4B and 8B while remaining comparable at 16B. These results show that the decoded-space objective is an independent design choice from the finite-rate coder and can materially affect downstream accuracy, particularly under tighter compression. They also indicate that Layer-RMS should be viewed as a query-agnostic, scale-normalized instantiation of our decoded-space objective rather than a universally optimal metric.

Effect of joint KV grouping. We study how the joint grouping of KV signals affects compression quality by varying the number of jointly represented heads G and layers P. Specifically, we vary $G \in \{1, 2, 4, 8\}$ while fixing $P = 1$, and vary $P \in \{1, 2, 4, 6, 9\}$ while fixing $G = 1$, under the same RVQ storage budgets of 8, 16, and 24 bytes. As shown in Fig. 3, increasing either the head-group size or the layer-group size generally improves StreamingBench performance at a fixed storage rate. The gain is particularly pronounced in the low-rate regime, while becoming smaller as the storage budget increases. This trend indicates that jointly encoding a larger set of KV sources exposes useful cross-head and cross-layer redundancy, allowing the codec to preserve more task-relevant information under a limited bit budget. At higher rates, where quantization error is already reduced, the benefit of additional joint modeling naturally becomes less significant.

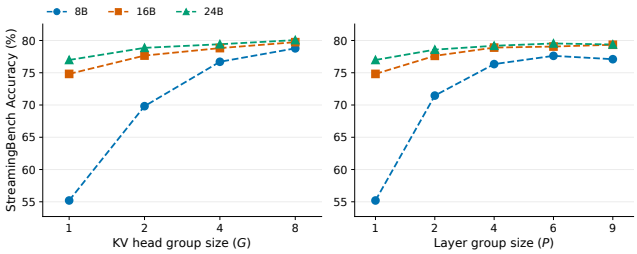


Figure 3: **Effect of joint KV grouping on StreamingBench.** We vary the number of jointly represented heads G with $P = 1$ (left), and the number of jointly represented layers P with $G = 1$ (right), under fixed RVQ storage budgets.

5 CONCLUSION

We introduced METRICKV, a query-agnostic visual-KV compression framework via decoded-space-geometry. A quadratic decoded-KV objective and decoder induce a latent pullback metric that measures finite-rate reconstruction error exactly. Absorbing this metric into the latent coordinates yields a Euclidean coding space, while the remaining coder-independent continuous subspace problem reduces to PCA in the distortion-transformed KV space. This separates the decoded-space objective, retained continuous subspace, and finite-rate coder. For streaming visual KV, we instantiate the framework with Layer-RMS normalization and progressive RVQ. The same Layer-RMS geometry improves a KVTC-style scalar codec across all tested rates, while a single progressive RVQ stream provides strong compression with limited downstream degradation. The codec remains complementary to memory retention and retrieval policies.

AI USE STATEMENT

In this work, we used generative AI tools to edit software code, draft parts of the paper, and improve readability. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed Abdelfattah, and Kai-Chiang Wu. Palu: Kv-cache compression with low-rank projection. In *International Conference on Learning Representations*, volume 2025, pp. 50222–50249, 2025.

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. Live: Learning video llm with streaming speech transcription at scale. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29083–29095. IEEE, 2025.

Yilong Chen, Xiang Bai, Zhibin Wang, Chengyu Bai, Yuhan Dai, and Ming Lu. Streamkv: Streaming video question-answering with segment-based kv cache retrieval and compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3120–3128, 2026.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. In *International Conference on Learning Representations*, volume 2025, pp. 42115–42127, 2025.

Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

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

Ian T Jolliffe. *Principal component analysis*. Springer, 2002.

Biing-Hwang Juang and A Gray. Multiple stage vector quantization for speech coding. In *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pp. 597–600. IEEE, 1982.

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

Junyan Li, Yang Zhang, Muhammad Yusuf Hassan, Talha Chafekar, Tianle Cai, Zhile Ren, Pengsheng Guo, Foroozan Karimzadeh, Chong Wang, and Chuang Gan. Commvq: Commutative vector quantization for kv cache compression. *arXiv preprint arXiv:2506.18879*, 2025.

Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12147–12151. IEEE, 2026.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18902–18913. IEEE, 2025.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24045–24055. IEEE, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding. *arXiv preprint arXiv:2604.02317*, 2026.

Konrad Staniszewski and Adrian Lancucki. Kv cache transform coding for compact storage in llm inference. In *International Conference on Learning Representations*, volume 2026, pp. 37768–37815, 2026.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. Vl-cache: Sparsity and modality-aware kv cache compression for vision-language model inference acceleration. In *International Conference on Learning Representations*, volume 2025, pp. 219–239, 2025.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Junbin Xiao, Jiajun Chen, Tianxiang Sun, Xun Yang, and Angela Yao. Mukv: Multi-grained kv cache compression for long streaming video question-answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11381–11391, 2026.

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. Streammem: Query-agnostic kv cache memory for streaming video understanding. *arXiv preprint arXiv:2508.15717*, 2025.

Yaoxin Yang, Peng Ye, Xudong Tan, Chongjun Tu, Maosen Zhao, Jia Hao, and Tao Chen. Revisiting multimodal kv cache compression: A frequency-domain-guided outlier-kv-aware approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39550–39560, 2026.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10807–10816, 2025.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, et al. Streamforest: Efficient online video understanding with persistent event memory. *Advances in Neural Information Processing Systems*, 38:75804–75835, 2026.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Haowei Zhang, Shudong Yang, Jinlan Fu, See Kiong Ng, and Xipeng Qiu. Hermes: Kv cache as hierarchical memory for efficient streaming video understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8411–8430, 2026.

Tianyi Zhang, Jonah Yi, Zhaozhuo Xu, and Anshumali Shrivastava. Kv cache is 1 bit per channel: Efficient large language model inference with coupled quantization. *Advances in Neural Information Processing Systems*, 37:3304–3331, 2024.

Appendix

The appendix is organized as follows:

- Appendix A: Proofs and canonical parameterization
- Appendix B: Detailed decoded-space reconstruction objectives
- Appendix C: RVQ details
- Appendix D: Additional experimental details
- Appendix E: Limitations

A Proofs and Canonical Parameterization

A.1 Proof of Theorem 1

Let

$$M = W^\top W, \qquad G = D^\top M D, \qquad E = G^{-1} D^\top M. \tag{18}$$

Because W is invertible and D has full column rank, $G \succ 0$. For a source vector x, let $z^\star = Ex$ and define the continuous projection residual

$$e = x - Dz^\star = x - DEx. \tag{19}$$

By the definition of E,

$$\begin{aligned} D^\top M e &= D^\top M x - D^\top M D E x \\ &= D^\top M x - G G^{-1} D^\top M x \\ &= 0. \end{aligned} \tag{20}$$

Thus, e is M-orthogonal to the column space of D. For any finite-rate latent reconstruction $\hat{z}$,

$$x - D\hat{z} = e + D(z^\star - \hat{z}). \tag{21}$$

The cross term in the squared M-norm vanishes by Eq. 20, so

$$\begin{aligned} \|W(x - D\hat{z})\|_2^2 &= \|We\|_2^2 + (z^\star - \hat{z})^\top D^\top M D (z^\star - \hat{z}) \\ &= \|W(x - Dz^\star)\|_2^2 + \|z^\star - \hat{z}\|_G^2 . \end{aligned} \tag{22}$$

Dividing by d proves Eq. 6. The first term is independent of $\hat{z}$; therefore, minimizing the decoded-space distortion over any candidate set is equivalent to minimizing the induced G-distance, proving Eq. 7.

The same identity also proves that $z^\star = Ex$ is the unique weighted least-squares coefficient for the fixed decoder D. For any $z' \in \mathbb{R}^r$,

$$\|W(x - Dz')\|_2^2 = \|W(x - DEx)\|_2^2 + \|Ex - z'\|_G^2 , \tag{23}$$

and the second term vanishes only at $z' = Ex$ because $G \succ 0$.

A.2 Canonical Euclidean Parameterization

Let $G = L^\top L$ and define the canonical coordinates as in Eq. 8:

$$E_c = LE, \qquad D_c = DL^{-1}. \tag{24}$$

Then

$$D_c^\top W^\top W D_c = L^{-\top} G L^{-1} = I_r, \tag{25}$$

and, using $E = G^{-1} D^\top W^\top W$,

$$\begin{aligned} E_c &= L G^{-1} D^\top W^\top W \\ &= L^{-\top} D^\top W^\top W \\ &= D_c^\top W^\top W. \end{aligned} \tag{26}$$

This proves Eq. 9. Moreover, $D_c E_c = DE$, so canonicalization changes only the latent coordinates, not the continuous decoded reconstruction.

Define $B = WD_c$. The first canonical condition gives $B^\top B = I_r$, while

$$D_c = W^{-1}B, \qquad E_c = B^\top W. \tag{27}$$

Conversely, any $B \in \mathbb{R}^{d\times r}$ with $B^\top B = I_r$ defines encoder and decoder maps of this form that satisfy Eq. 9. Hence Eq. 11 characterizes the complete canonical Euclidean family.

A.3 Proof of Proposition 1

Let $\bar{x} = Wx$ and consider a canonical representation parameterized by B. By Eq. 11,

$$DEx = W^{-1}BB^\top \bar{x}. \tag{28}$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\mathcal{D}_W(x, DEx)\right] &= \frac{1}{d}\mathbb{E}\left[\left\|(I - BB^\top)\bar{x}\right\|_2^2\right] \\ &= \frac{1}{d}\operatorname{tr}\left((I - BB^\top)\Sigma\right) \\ &= \frac{1}{d}\left[\operatorname{tr}(\Sigma) - \operatorname{tr}(B^\top \Sigma B)\right],\end{aligned} \tag{29}$$

where $\Sigma = \mathbb{E}[\bar{x}\bar{x}^\top]$ is the covariance matrix used in Sec. 3.3. Since $\operatorname{tr}(\Sigma)$ is independent of B, minimizing the expected continuous reconstruction error is equivalent to

$$\max_{B^\top B = I_r} \operatorname{tr}(B^\top \Sigma B). \tag{30}$$

By the Ky Fan maximum principle (Fan, 1949), the maximum is attained when the columns of B span an eigenspace associated with the largest r eigenvalues of Σ. Thus, one minimizer is $B = U_r$, proving Proposition 1.

B Detailed Decoded-Space Reconstruction Objectives

Section 3.2 defines a general decoded-space quadratic distortion through a fixed invertible transform W. The experiments use two instances of this objective: Raw MSE and Layer-RMS. They differ only in W; the subsequent pullback construction, canonicalization, subspace selection, and finite-rate coder are unchanged.

Raw MSE. No rescaling is applied:

$$W_{\text{raw}} = I_d. \tag{31}$$

This recovers ordinary mean squared reconstruction error.

Layer-RMS. For the joint KV vector in Eq. 1, define one block for each layer $\ell \in b$:

$$x_\ell = \operatorname{Concat}_{h\in g}\left[K_{\ell,h}, V_{\ell,h}\right] \in \mathbb{R}^{2Gd_h}. \tag{32}$$

Its population RMS is

$$s_\ell = \sqrt{\frac{1}{2Gd_h}\mathbb{E}\left[\|x_\ell\|_2^2\right]}. \tag{33}$$

For a bucket containing P layers, Layer-RMS uses

$$W_{\text{RMS}} = \operatorname{diag}\left(\frac{1}{s_{\ell_1}} I_{2Gd_h}, \ldots, \frac{1}{s_{\ell_P}} I_{2Gd_h}\right), \tag{34}$$

which yields

$$\mathcal{D}_{W_{\text{RMS}}}(x, \hat{x}) = \frac{1}{d}\sum_{\ell\in b}\frac{\|x_\ell - \hat{x}_\ell\|_2^2}{s_\ell^2}. \tag{35}$$

Thus, each layer contributes reconstruction error relative to its typical activation magnitude, while the relative geometry among heads and between keys and values within the same layer is preserved. If one layer block and its population statistics are rescaled by a nonzero scalar, its normalized contribution in Eq. 35 is unchanged. This is the layer-wise scale invariance referred to in the main text; it is a property of the decoded-space objective rather than a separate representation construction.

C RVQ DETAILS

C.1 STAGE-WISE RVQ CONSISTENCY

For the final maps in Eq. 14, $D^\top W^\top W D = I_r$. Hence the finite-rate term in Eq. 10 is ordinary Euclidean distance in the latent space. At RVQ stage s, let $\hat{z}^{(s-1)}$ be the reconstruction formed by the first $s-1$ codewords. Each candidate next reconstruction is

$$\hat{z}^{(s-1)} + c, \qquad c \in C_{b,g,s}. \tag{36}$$

Applying Theorem 1 with $z = Ex$ gives

$$\begin{aligned}\operatorname*{argmin}_{c \in C_{b,g,s}} \left\| z - (\hat{z}^{(s-1)} + c) \right\|_2^2 &= \operatorname*{argmin}_{c \in C_{b,g,s}} \mathcal{D}_W \left(x, D(\hat{z}^{(s-1)} + c) \right) \\ &= \operatorname*{argmin}_{c \in C_{b,g,s}} \left\| r^{(s-1)} - c \right\|_2^2 .\end{aligned} \tag{37}$$

Therefore, each greedy RVQ assignment ranks the codewords available at that stage exactly as the target decoded-KV distortion does. This is a local stage-wise consistency statement; it does not imply that greedy RVQ is globally optimal over all combinations of codewords across stages.

C.2 DETAILED CODEC CONSTRUCTION

The codec is constructed in three steps. The continuous representation is determined analytically before any finite-rate codebook optimization and remains fixed thereafter.

A. Analytic representation construction. For each layer-group–head-group bucket (b, g), we estimate the Layer-RMS statistics from the unlabeled fitting KV, form $\bar{x} = Wx$, and estimate

$$\Sigma = \mathbb{E}[\bar{x}\bar{x}^\top]. \tag{38}$$

We compute its leading r-dimensional eigenspace U_r and construct $E = U_r^\top W$ and $D = W^{-1} U_r$ according to Eq. 14. No gradient-based optimization is used for E or D.

B. Residual k-means initialization. All fitting vectors are mapped into the fixed canonical latent $z = Ex$. The RVQ codebooks are initialized sequentially. At stage s, k-means is applied to the current residuals $r^{(s-1)}$; the selected centroid is subtracted before fitting stage $s+1$. This produces an ordered sequence of residual codebooks in the same Euclidean latent geometry used at inference time.

C. Multi-prefix codebook refinement. After initialization, only the codebooks $\{C_s\}_{s=1}^{S_{\max}}$ are updated. Let $\mathcal{S}$ denote the supported prefix lengths. For each minibatch, we sample $S \sim \mathrm{Unif}(\mathcal{S})$ and use

$$\begin{aligned}\mathcal{L}_{\mathrm{rec}}(S) = \mathcal{D}_W \left(x, DQ_S(Ex) \right) \\ + \mathbf{1}_{S<S_{\max}} \lambda_{\max} \mathcal{D}_W \left(x, DQ_{S_{\max}}(Ex) \right).\end{aligned} \tag{39}$$

The first term optimizes the sampled operating point, while the auxiliary full-prefix term anchors the highest-rate reconstruction when a shorter prefix is sampled. Nearest-codeword indices are treated as discrete selections: no gradient is taken through the argmin operation, and gradients update only the selected codewords. Because E and D are fixed, no straight-through estimator is needed for the representation transform.

We additionally use the stage-wise centroid objective

$$\mathcal{L}_{\mathrm{cb}} = \frac{1}{S_{\max}} \sum_{s=1}^{S_{\max}} \left\| C_s[a^{(s)}] - \mathrm{stopgrad}\left(r^{(s-1)} \right) \right\|_2^2 , \tag{40}$$

and optimize

$$\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{cb}} \mathcal{L}_{\mathrm{cb}}. \tag{41}$$

Table 5: Controlled KV-compression results on Qwen3.5 4B under the same fixed-readout protocol as Table 1.

Method	Storage	Compression	OVO-Bench			StreamingBench
			Real-time	Backward w/o HLD	OVO Avg.	
FP16	1024	1.00×	77.89	64.58	71.24	75.62
METRICKV (S256)	8	128.00×	74.01	62.56	**68.29**	**71.10**
METRICKV (S512)	16	64.00×	76.43	64.25	70.34	73.78
METRICKV (S1024)	32	32.00×	78.26	64.58	71.42	**75.82**
KVTC (Staniszewski & Lancucki, 2026)	8	128.00×	71.51	59.02	65.26	69.42
KVTC (Staniszewski & Lancucki, 2026)	16	64.00×	76.87	64.08	70.47	73.86
KVTC (Staniszewski & Lancucki, 2026)	32	32.00×	78.78	63.91	71.34	75.62
KVTC + Layer-RMS	8	128.00×	72.69	60.03	66.36	70.62
KVTC + Layer-RMS	16	64.00×	77.34	65.09	**71.22**	**73.98**
KVTC + Layer-RMS	32	32.00×	78.52	64.41	**71.47**	75.70

The centroid term encourages the selected codeword at each stage to match the residual presented to that stage. We set $\lambda_{\max} = 0.25$ and $\lambda_{\mathrm{cb}} = 0.05$. Since only the codebooks are refined, this optimization cannot change the canonical latent geometry established in Sec. 3.3; all nearest-codeword decisions remain Euclidean realizations of the decoded-distortion pullback metric. No future query, answer label, or gradient through the frozen MLLM is used during codec construction.

For the default configuration in the main experiments, $M = 16$ and the supported RVQ prefixes are

$$\mathcal{S} = \{192, 384, 768\}, \qquad S_{\max} = 768. \tag{42}$$

With the default grouping of six layers and four KV heads, Eq. 17 gives 4, 8, and 16 bytes per original head/layer/token for these three prefixes, respectively.

C.3 OPTIMIZATION DETAILS.

Each codec bucket uses 500,000 aligned KV vectors. We reserve 5% of the vectors for validation, resulting in 475,000 training vectors and 25,000 validation vectors per bucket. The codebooks are optimized for two epochs using AdamW with a constant learning rate of 5×10^{-5}. We use a logical batch size of 256 vectors and clip the global gradient norm to 1.0.

D ADDITIONAL EXPERIMENTAL DETAILS

D.1 RESULTS ON QWEN3.5 4B

To examine whether the observed compression behavior transfers to another backbone, we repeat the controlled fixed-readout evaluation on Qwen3.5 4B (Qwen Team, 2026), following the same protocol as Table 1. We set $P = 4$ and $G = 4$ for Qwen3.5. Table 5 compares METRICKV with KVTC and its Layer-RMS variant at matched storage budgets.

The results show a similar rate–accuracy trade-off on Qwen3.5 4B. At 32B, METRICKV achieves 71.42 on OVO-Bench and 75.82 on StreamingBench, essentially matching the FP16 baseline despite 32× compression. At the most aggressive 8B operating point (128×), it outperforms vanilla KVTC by 3.03 points on OVO-Bench and 1.68 points on StreamingBench. Moreover, replacing Raw MSE with Layer-RMS consistently improves KVTC on both benchmarks across all three storage budgets, with the largest gains at 8B. This provides an additional cross-backbone validation that the choice of decoded-space objective remains important, particularly under severe compression.

D.2 EMPIRICAL VALIDATION OF FINITE-RATE REPRESENTATION CONSISTENCY

We empirically validate the two distinct roles separated in Sec. 3.3. First, we reparameterize a fixed canonical codec while preserving its decoded candidate reconstructions, and test whether finite-rate assignments remain invariant to the latent coordinates. Second, we keep the latent metric canonical and vary only the retained subspace, isolating the coder-independent continuous projection objective in Proposition 1.

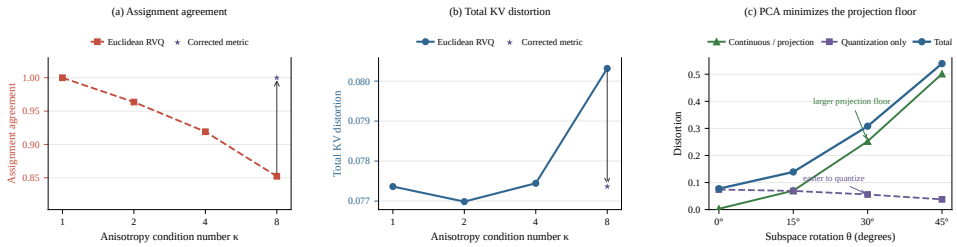


Figure 4: **Finite-rate coordinate dependence and continuous subspace selection. (a–b)** We anisotropically reparameterize a fixed canonical codec while transforming the encoder, decoder, and codewords so that every corresponding candidate has the same decoded reconstruction. If Euclidean assignment is incorrectly retained, agreement with the canonical RVQ decisions decreases as the anisotropy grows and decoded-KV distortion becomes coordinate-dependent. Applying the induced pullback metric at $\kappa = 8$ recovers the canonical assignments and distortion. **(c)** Keeping the latent metric canonical, rotating the retained subspace away from the principal eigenspace increases the continuous projection term. Although the quantization-only term decreases, the total distortion increases in this diagnostic.

Ignoring the pullback metric makes finite-rate coding coordinate-dependent. Starting from the trained canonical codec, let $A_\kappa = \mathrm{diag}(a_1, \ldots, a_r)$, where $a_j = \kappa^{(j-1)/(r-1)}$ and $\kappa \in \{1, 2, 4, 8\}$. We reparameterize

$$E_\kappa = A_\kappa E, \qquad D_\kappa = D A_\kappa^{-1}, \qquad c_\kappa = A_\kappa c. \tag{43}$$

For every latent vector and every correspondingly transformed candidate, the decoded output is unchanged because $D_\kappa A_\kappa = D$. However, if Euclidean distance is incorrectly retained after the reparameterization, a stage assignment becomes

$$a_\kappa^{(s)} = \operatorname*{argmin}_m \left\| A_\kappa \left(r^{(s-1)} - c_{s,m} \right) \right\|_2^2. \tag{44}$$

This generally differs from the canonical Euclidean decision even though the underlying decoded candidates are unchanged.

Theorem 1 specifies the metric that must accompany the reparameterized decoder. Because the original canonical metric is I_r,

$$G_\kappa = D_\kappa^\top W^\top W D_\kappa = A_\kappa^{-\top} A_\kappa^{-1}. \tag{45}$$

Consequently, for any residual difference u,

$$\|A_\kappa u\|_{G_\kappa}^2 = \|u\|_2^2, \tag{46}$$

so the corrected metric exactly preserves the original candidate ordering.

Figure 4(a) shows that, when this correction is omitted, assignment agreement with the canonical codec decreases steadily as the reparameterization becomes more anisotropic. Panel (b) shows the corresponding coordinate dependence of decoded-KV distortion: the effect need not be monotonic for mild reparameterizations, but becomes pronounced at $\kappa = 8$. Using G_κ at the same κ restores the canonical assignments and returns the distortion to the canonical reference. Thus, the observed changes arise from the finite-rate assignment geometry rather than from any change in the continuous reconstruction or decoded candidate set.

PCA minimizes the coder-independent projection floor. We next keep the latent metric canonical and rotate the leading eigenspace U_r toward an orthogonal r-dimensional eigenspace V_r:

$$B_\theta = U_r \cos\theta + V_r \sin\theta, \tag{47}$$

where $B_\theta^\top B_\theta = I_r$ for every θ. From 0° to 45°, the continuous distortion increases from 0.003158 to 0.501398, while the quantization-only term for the RVQ prefix decreases from 0.074087 to 0.037831. The total distortion nevertheless increases from 0.077246 to 0.539229.

This result directly illustrates the decomposition in Eq. 12. PCA minimizes the coder-independent continuous projection term, while the finite-rate term depends separately on the coder, rate, and

retained subspace and can therefore vary in the opposite direction. In this diagnostic, the reduction in quantization error is insufficient to offset the larger projection floor, yielding higher total distortion away from the principal subspace.

D.3 Streaming Protocol and Fixed Readout

All videos are sampled at 1 fps and processed causally. Visual KV is generated from encoding chunks containing five consecutive sampled frames. For the controlled representation-level comparison in Table 1, each query uses four externally retrieved historical frames and the four most recent frames available before the query. Historical retrieval follows the SigLIP-based (Zhai et al., 2023) external index used by ReKV (Di et al., 2025).

The selected frame identities are determined independently of the KV codec and then reused unchanged for FP16 and every compression baseline. Accordingly, all methods receive the same visual evidence, frame ordering, positional treatment, and answer-time visual-token budget. Compression is applied to the stored visual KV states; at query time, only the selected compressed states are decoded and passed to the frozen MLLM. This protocol isolates representation quality from evidence-selection quality.

For the system-level results in Table 2, we instead preserve each streaming system's native memory policy. ReKV + MetricKV retains the same historical selections as ReKV and changes only their KV representation, whereas SimpleStream + MetricKV keeps the original four-frame recent window and compresses the retained KV states.

E Limitations

MetricKV compresses the representation of retained visual tokens; it does not itself decide which historical evidence should be retained or retrieved, and it does not reduce the number of decoded visual tokens processed by attention once a readout has been chosen. Its end-to-end benefit therefore depends on the surrounding memory policy and on the backbone's sensitivity to KV reconstruction error.

The pullback consistency in Theorem 1 is exact for the specified quadratic decoded-space objective $\mathcal{D}_W$, not for downstream question-answering accuracy or attention outputs under every possible future query. Proposition 1 selects the subspace that minimizes the coder-independent continuous projection term; it is not a claim that PCA jointly minimizes the final finite-rate distortion for every coder, rate, or fitted codebook.

The codec is model-specific: normalization statistics, transforms, and codebooks are fitted for each backbone and layer-group–head-group bucket. Static codec parameters are amortized only over sufficiently long streams or repeated sessions using the same codec. Finally, the current fitting and evaluation benchmarks cover web-video style data and do not establish robustness for substantially different streaming domains such as egocentric, embodied, medical, or safety-critical video.