CORAL-KV: COMPACT CROSS-LAYER PREDICTION AND RESIDUAL CODING FOR AUTOREGRESSIVE VIDEO KV CACHES

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive video diffusion models maintain a growing attention key–value (KV) cache, making memory a key bottleneck for long video generation. Recently, clustering-based quantization has emerged as an effective method to compress video KV caches by grouping similar key and value vectors and quantizing their residuals. However, when later frames attend to the compressed history, small reconstruction errors can compound as the video grows. Beyond the within-layer similarity exploited by clustering, we find strong dependencies across cache layers in video models. This motivates CORAL-KV—**C**alibrated **O**ffline, **R**esidual-coded, and **A**dapted **L**ocally—a KV-cache compression framework that combines reusable cross-layer predictors with low-bit residual quantization. We use reconstructed cache states to predict keys and values in subsequent layers, then quantize the resulting prediction residuals. We construct compact block-diagonal-plus-low-rank (block-DPLR) predictors: block-diagonal components model channel relationships within each attention head, while a global low-rank term models dependencies between heads. CORAL-KV exposes a flexible fidelity–memory trade-off: it can reuse its predictors across videos or refit lightweight block-diagonal predictors to the current video for higher fidelity. Across extensive evaluations on both HY-WorldPlay and MAGI-1, our reusable configuration improves PSNR over QVG-4 by 0.32 dB and 4.95 dB, respectively, while using 48% and 37% less cache memory. Against QVG-Pro, our video-adapted configuration improves MAGI-1 PSNR by 3.77 dB while using 33.7% less cache memory. CORAL-KV also maintains higher reconstruction fidelity over extended generation and reduces temporal discontinuities while retaining comparable generation latency.

1 INTRODUCTION

Recent advances in video generation point toward a powerful new paradigm: learned simulators that can model and reason about the dynamics of the physical world (Polyak et al., 2024; Kong et al., 2024; Team Wan et al., 2025; Google DeepMind, 2025; Wiedemer et al., 2025). Beyond content creation, such systems could support interactive environments, embodied agents, and general-purpose visual reasoning (He et al., 2025; Xiao et al., 2025). Autoregressive video models are particularly well suited to realizing this promise: by generating video incrementally while conditioning each new segment on the preceding history, they support causal interaction and long-horizon coherence (Huang et al., 2025a; Sand.ai et al., 2025; Yang et al., 2025b).

Despite their promise, autoregressive video models face a severe memory bottleneck: the attention key–value (KV) cache. This cache grows with video duration, spatial resolution, and model depth, and can quickly dominate GPU memory (Li et al., 2026; Samuel et al., 2026). Consequently, memory constraints limit both where these models can be used and how much history they can retain, even though longer context is critical for preserving identity, scene geometry, and motion over long rollouts (Hong et al., 2025; Xiao et al., 2025).

Prior work has shown that video KV caches can be compressed by exploiting spatiotemporal redundancy among video tokens (Xi et al., 2025; Yang et al., 2025c). In particular, Quant VideoGen

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.

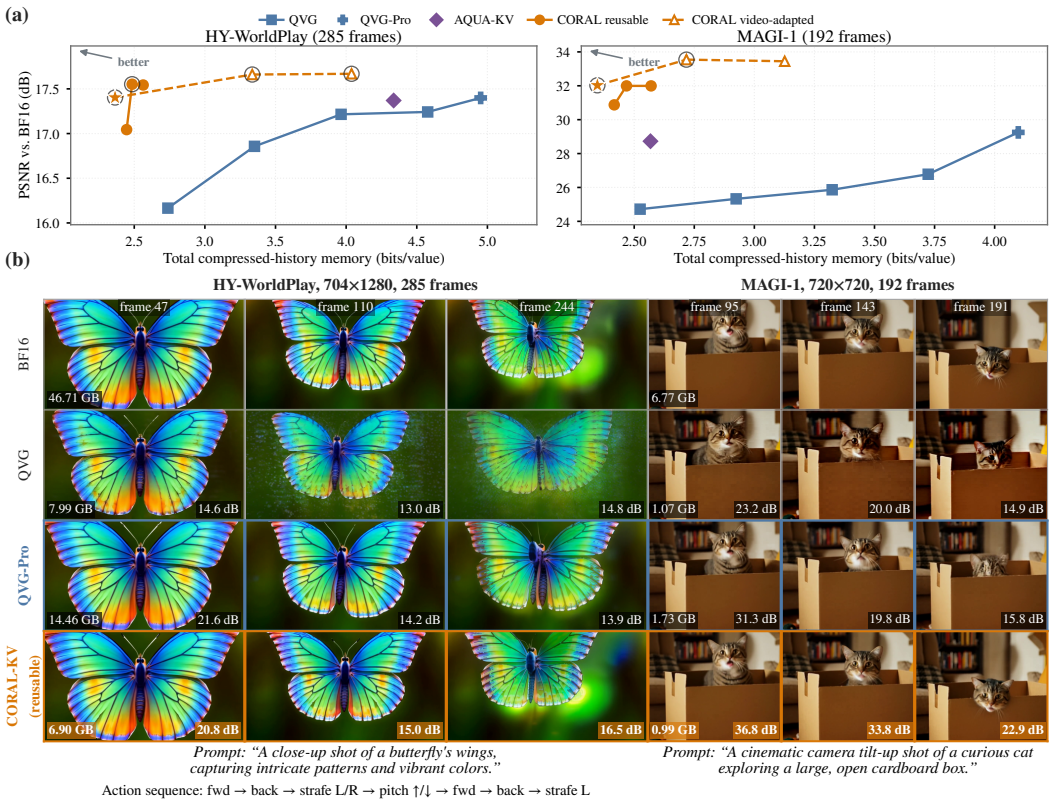


Figure 1: **CORAL-KV improves the memory–quality frontier on both models.** (a) PSNR versus total compressed-history memory across 32 prompts per model, using the accounting in Table 1; stars mark our selected reusable codec. (b) Qualitative comparisons of BF16, QVG, QVG-Pro, and reusable CORAL-KV on matched frames.

(QVG) realizes this idea by separately clustering similar key vectors and similar value vectors within each layer, representing each cluster with a centroid, and quantizing the residuals to low-bit precision (Xi et al., 2026). Over long generation horizons, however, we observe that small errors early in generation can propagate through later frames, producing substantial reconstruction drift (Figure 1).

Instead of clustering similar vectors within each cache layer, we find strong cross-layer dependencies in video KV caches. These dependencies hold across two distinct video architectures (Figure 2). AQUA-KV observed related dependencies in LLM KV caches and used dense linear predictors with residual quantization (Shutova et al., 2025). Directly transferring this approach to video models, however, introduces new bottlenecks: dense predictor memory scales quadratically with the KV hidden dimension, and applying these predictors at every layer adds substantial inference overhead. We address these bottlenecks with CORAL-KV, a practical KV-cache compression framework that combines compact block-diagonal-plus-low-rank (block-DPLR) predictors with low-bit residual quantization. In summary, our contributions are as follows:

- We introduce CORAL-KV, a KV-cache compression framework that makes cross-layer prediction practical with block-DPLR predictors that preserve within- and cross-head information using $13.1\times$ and $3.0\times$ fewer matrix parameters than dense prediction on HY-WorldPlay and MAGI-1, respectively. They can be calibrated once offline for reuse across videos or adapted online by refitting only block-diagonal components per cache block for higher fidelity.
- We co-design prediction and residual quantization by calibrating predictors on reconstructed cache states and tailoring blockwise Hadamard transforms and 2-bit vector quantization to their residuals. Direct factorized reconstruction avoids dense predictor matrices, reducing memory and inference overhead.
- Through extensive evaluations on two video generators, we show that CORAL-KV improves visual fidelity and temporal stability over matched baselines at comparable generation latency.

2 RELATED WORK

We focus here on the three areas most directly related to CORAL-KV: video-cache compression and attention reduction, LLM KV compression, and low-bit quantization. We provide an expanded literature map covering these topics, autoregressive video generation, and evaluation protocols in Appendix B.

Video-cache compression and attention reduction. We directly compare CORAL-KV to QVG, which fits per-head centroid codebooks online for each cache block and progressively quantizes deviations from them (Xi et al., 2026). Other video methods prune or compact tokens, sparsify attention, replace dense attention state with low-rank representations, or absorb history into online parameters (Xi et al., 2025; Yang et al., 2025c; Ji et al., 2026b; Zhao et al., 2026; Li et al., 2026; Yesiltepe et al., 2026; Fu et al., 2026). These approaches change which historical tokens remain accessible or how attention uses them; CORAL-KV retains every position while reducing the memory needed to store its KV values.

LLM KV compression and cross-layer prediction. Prior LLM methods quantize or correct cached state to reduce its memory footprint (Liu et al., 2024b; Hooper et al., 2024; He et al., 2024; Kang et al., 2024). We find cross-layer predictability in video caches and show that predictors calibrated offline remain effective across videos. We exploit these findings with compact block-DPLR predictors and low-bit residual coding. AQUA-KV reports related cross-layer predictability in LLM caches and uses dense predictors with residual quantization (Shutova et al., 2025). Dense maps scale quadratically with channel width, making them costly for wide video caches; our factorized predictors reduce memory and inference cost.

Quantization and residual coding. Prior quantization methods use orthogonal rotations to distribute outliers and improve low-bit coding (Ashkboos et al., 2024; Liu et al., 2025). We apply fixed Hadamard transforms to cross-layer prediction residuals and choose how many residual channels each transform mixes using our quantization and video-quality ablations (Figure 5; Appendix H.2). We then encode the transformed residuals with the fixed, Gaussian-tuned 2-bit EDEN-2 vector codebook (Vargaftik et al., 2022; Malinovskii et al., 2025).

3 PROBLEM SETTING AND MOTIVATION

Video KV caches are extremely memory-intensive. In an autoregressive video model with L attention layers, KV channel width C, and T retained patch tokens, storing both keys and values in BF16 requires $2LTC \times 16$ bits. For a 285-frame, near-720p HY-WorldPlay video, retaining the full BF16 KV cache requires 46.71 GB (Table 1). This highlights the critical need to compress cache memory to enable long-horizon video generation.

Quantizing cache blocks can compound reconstruction errors. One way to reduce video KV-cache memory is to quantize previously generated keys and values. During generation, KV states for new visual tokens accumulate in a *cache block*. When the block reaches its token limit, it is compressed and retained as history while generation continues with a new block. QVG fits centroid codebooks *online* for each filled block, clustering similar keys and values within each layer before quantizing their deviations from the centroids (Xi et al., 2026).

Although this can substantially reduce cache size, we observe that it introduces small reconstruction errors early in the generated video, which can become magnified as the video grows longer, manifesting in our qualitative examples as oversmoothing, blended structures, and geometric drift (Figure 1; see Appendix Figures 10–13 for additional examples). Because this conversion occurs at a discrete boundary, the next generated frame is the first frame that attends to the newly compressed history. This transition can cause an abrupt quality drop and a visible temporal discontinuity (Figure 2(c)). We call this effect quantization-boundary discontinuity (QBD) and define its measurement in Section 5.

Cross-layer structure and reusable predictors. Rather than clustering keys and values within each cache layer, we examine dependencies across layers. In HY-WorldPlay and MAGI-1, we train linear maps to predict a layer's keys and values from preceding-layer cache states (Figure 2(a,b)). We also find that predictors fitted on one set of videos remain effective on held-out videos, so a single predictor bank can be calibrated *offline* and shared across videos. Let K_ℓ and V_ℓ denote the unquan-

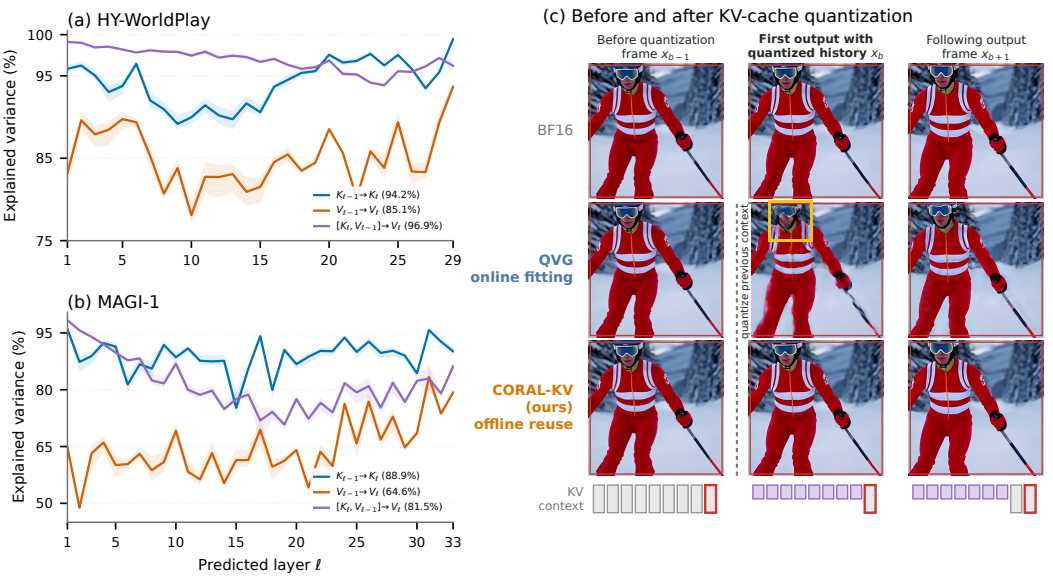


Figure 2: **Motivation for cross-layer prediction.** (a–b) Explained variance across depth for three tokenwise linear predictors of unquantized cache states. Lines and shading show the mean and range across three cache dumps. (c) QVG fits video-specific centroids online when a cache block reaches its token limit, whereas CORAL-KV reuses offline-calibrated predictors. The yellow box highlights facial distortion in QVG's first frame generated from compressed history; visible degradation also appears in the clothing and ski pole.

tized keys and values at layer ℓ. We compare three tokenwise linear-map structures: keys to keys ($K_{\ell-1} \to K_\ell$), values to values ($V_{\ell-1} \to V_\ell$), and keys and values to values ($[K_\ell, V_{\ell-1}] \to V_\ell$). For the key-to-key map, AQUA-KV reports about 80% explained variance on Llama-3.2-3B (Shutova et al., 2025), versus 94% on HY-WorldPlay and 89% on MAGI-1 in our held-out video probes. Adding current-layer keys improves value prediction on both models. This motivates reconstructing keys first, then using them with preceding-layer values to predict current-layer values. Because preceding-layer keys already predict current keys well, we omit preceding-layer values from the key predictor to avoid doubling its input width and reduce predictor memory. We do not have a formal explanation for this linear predictability. A plausible intuition is that adjacent layers process the same token history and retain shared structure; the held-out curves and cross-video reuse are consistent with this view but do not establish it (Appendix A).

4 CORAL-KV

4.1 Reconstruction overview

Figure 3 illustrates how CORAL-KV reconstructs each layer of a compressed cache block. Once a block is full, we encode its KV states and retain quantized codes and scales. When later chunks attend to that block, we reconstruct its keys and values layer by layer. At layer $\ell > 0$, a tokenwise linear map P_ℓ^K predicts keys from reconstructed keys at layer $\ell - 1$. A second map P_ℓ^V predicts values from reconstructed keys at layer ℓ and reconstructed values at layer $\ell - 1$. Section 4.2 gives their compact parameterization. We correct both predictions with quantized residuals; let $\mathcal{Q}_\mathrm{R}$ denote residual quantization followed by dequantization:

$$\text{predicted keys: } \widetilde{K}_\ell = P_\ell^K(\widehat{K}_{\ell-1}), \qquad \text{reconstructed keys: } \widehat{K}_\ell = \widetilde{K}_\ell + \mathcal{Q}_\mathrm{R}(K_\ell - \widetilde{K}_\ell),$$
$$\text{predicted values: } \widetilde{V}_\ell = P_\ell^V([\widehat{K}_\ell, \widehat{V}_{\ell-1}]), \qquad \text{reconstructed values: } \widehat{V}_\ell = \widetilde{V}_\ell + \mathcal{Q}_\mathrm{R}(V_\ell - \widetilde{V}_\ell). \tag{1}$$

Here, brackets concatenate channels. The predictors do not receive original keys or values as inputs; they operate only on reconstructed tensors available to the decoder. Because layer zero has no preceding layer, we quantize its keys and values directly. In our ablations, we find that INT4 is sufficient

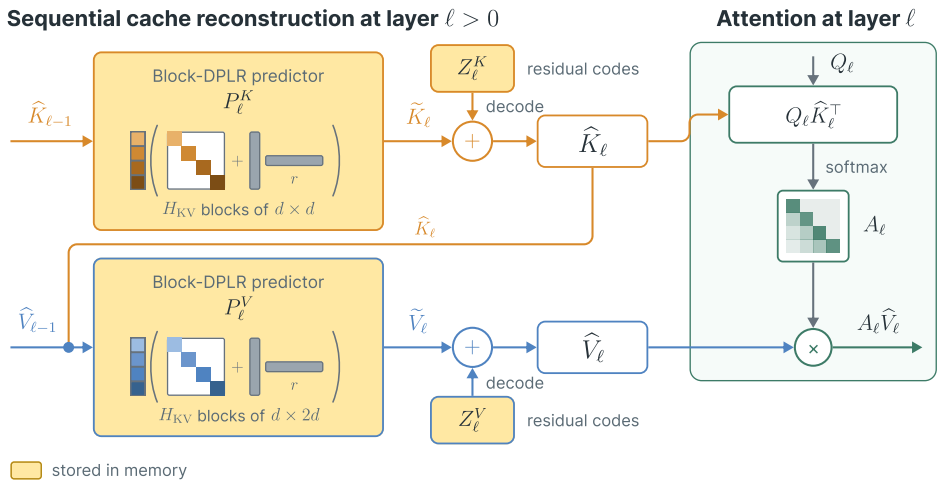


Figure 3: **CORAL-KV reconstruction.** At layer $\ell > 0$, the key predictor maps $\widehat{K}_{\ell-1}$ to $\widetilde{K}_\ell$; adding a decoded residual gives $\widehat{K}_\ell$. This reconstructed key feeds both attention and the value predictor. The value predictor also receives $\widehat{V}_{\ell-1}$; its output plus a decoded residual gives $\widehat{V}_\ell$. Both predictors use a block-diagonal component and a rank-r low-rank component.

Algorithm 1 Block-DPLR calibration per layer

Require: Inputs $X_\ell^K = \widehat{K}_{\ell-1}$ and $X_\ell^V = [\widehat{K}_\ell, \widehat{V}_{\ell-1}]$; targets K_ℓ, V_ℓ; rank r; three fitting rounds
Ensure: $P_\ell^S(x) = D_\ell^S x + A_\ell^S (B_\ell^S x)$ for $S \in \{K, V\}$
Initialize $(A_\ell^K B_\ell^K)^{(0)} = 0$ and $(A_\ell^V B_\ell^V)^{(0)} = 0$
for $t = 1, 2, 3$ **do**
  // Fit the block-diagonal maps to the remaining error
  $R_D^K \leftarrow K_\ell - (A_\ell^K B_\ell^K)^{(t-1)} X_\ell^K$; $R_D^V \leftarrow V_\ell - (A_\ell^V B_\ell^V)^{(t-1)} X_\ell^V$
  $D_\ell^{K,(t)} \leftarrow \arg\min_{D \in \mathcal{D}^K} \|D X_\ell^K - R_D^K\|_F^2$; $D_\ell^{V,(t)} \leftarrow \arg\min_{D \in \mathcal{D}^V} \|D X_\ell^V - R_D^V\|_F^2$
  // Fit the low-rank maps to the remaining error
  $R_{AB}^K \leftarrow K_\ell - D_\ell^{K,(t)} X_\ell^K$; $R_{AB}^V \leftarrow V_\ell - D_\ell^{V,(t)} X_\ell^V$
  $(A_\ell^{S,(t)}, B_\ell^{S,(t)}) \leftarrow \arg\min_{\mathrm{rank}(AB) \le r} \|AB X_\ell^S - R_{AB}^S\|_F^2$ for $S \in \{K, V\}$
end for
return P_ℓ^K, P_ℓ^V

to preserve video fidelity for layer zero on both models (Appendix H.1; Figure 7). Appendix C.1 gives the full encoder–decoder specification.

4.2 Compact predictors and residual quantization

Directly adopting dense cross-layer predictors as in AQUA-KV can erase the memory savings they enable. For KV channel width C, a dense key predictor contains C^2 parameters and a dense value predictor contains $2C^2$, for a total of $3C^2$ per layer. For instance, HY-WorldPlay has $C = 6144$, so storing these predictors in BF16 requires 6.57 GB across its predicted layers. In our evaluated setting, the predictors alone require nearly twice as much memory as the compressed cache, while applying them at every layer increases generation time by 71.7% relative to BF16 (Table 1). A practical KV compressor therefore requires a predictor that retains dense-map expressiveness without quadratic memory and computation.

We propose a block-diagonal-plus-low-rank (block-DPLR) predictor with a block-diagonal component D_ℓ and a rank-r component $A_\ell B_\ell$. Let H_{KV} denote the number of KV attention heads and $d = C/H_{\mathrm{KV}}$ their width. For keys, each head-aligned block $D_{\ell,h}^K \in \mathbb{R}^{d \times d}$ maps the corresponding preceding-layer key head, while $A_\ell^K \in \mathbb{R}^{C \times r}$ and $B_\ell^K \in \mathbb{R}^{r \times C}$ incorporate information across heads. For values, each $D_{\ell,h}^V \in \mathbb{R}^{d \times 2d}$ maps the corresponding current-layer key and preceding-

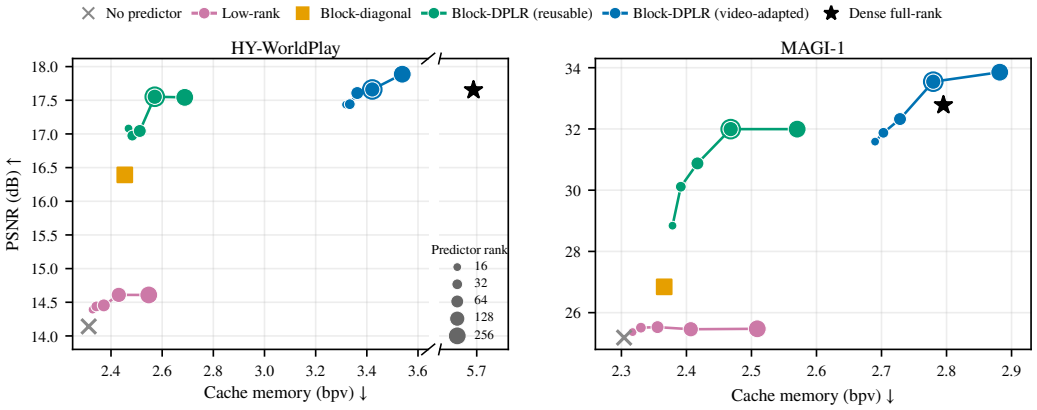


Figure 4: **Predictor-structure ablation.** Block-DPLR outperforms block-diagonal and parameter-matched low-rank predictors. Reusable block-DPLR shares an offline predictor bank across videos. Video-adapted block-DPLR keeps its low-rank factors fixed and refits block-diagonal terms for each filled cache block, improving fidelity at higher memory cost. The dense control is also fitted offline.

layer value heads, while $A_\ell^V \in \mathbb{R}^{C\times r}$ and $B_\ell^V \in \mathbb{R}^{r\times 2C}$ provide the cross-head component. Aligning the blocks with attention heads permits independent headwise fits and head-batched application.

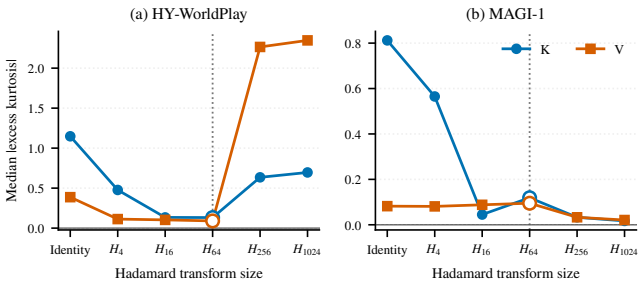
Fitting these components independently can cause them to learn redundant information. We instead use alternating block minimization (Tseng, 2001) (Algorithm 1). Each update fits one component to the target residual after subtracting the current prediction of the other component, so the two components model complementary information. We write $\mathcal{D}^K$ and $\mathcal{D}^V$ for the corresponding head-aligned block-diagonal matrix families.

The key and value predictors use $H_{\mathrm{KV}}d^2+2Cr$ and $2H_{\mathrm{KV}}d^2+3Cr$ matrix parameters, respectively. At rank $r = 128$, block-DPLR uses $13.1\times$ fewer matrix parameters than dense prediction on HY-WorldPlay and $3.0\times$ fewer on MAGI-1. At inference, we apply $D_\ell x$ in head batches and compute the low-rank term $A_\ell(B_\ell x)$ without forming a dense predictor. We develop fused GPU kernels that decode residuals, apply these factors, and write reconstructed states directly into the attention layout. Appendix C.2 provides implementation details and latency breakdowns.

Predictor structure. The alternating fit calibrates one key and one value predictor per layer from cache tokens pooled across 32 diverse MovieGen Extended videos (Polyak et al., 2024); *reusable* CORAL-KV shares this offline-calibrated bank across videos. Results with 8–16 calibration videos approach those with 32 (Table 9). *Video-adapted* CORAL-KV keeps the offline low-rank factors but refits block-diagonal terms online for each filled cache block of the current video. We compare offline block-diagonal, low-rank, block-DPLR, and dense predictors across parameter budgets, alongside a no-predictor control. Figure 4 shows that block-DPLR outperforms block-diagonal and parameter-matched low-rank predictors on both models. The offline dense predictor gains only 0.10 and 0.79 dB over reusable rank-128 block-DPLR on HY-WorldPlay and MAGI-1, respectively, while requiring substantially more predictor memory.

This per-block refitting improves fidelity at a memory cost. Online low-rank refitting gives similar fidelity, but its global update takes $2.6\times$ as long to fit as the parallel headwise block-diagonal update. We therefore keep the offline low-rank factors fixed in the video-adapted configuration, which gives the highest quality in Figure 4. Appendix H.1 reports the full ablations and confidence intervals.

Quantizing residuals. To preserve the predictor's reconstruction gains within a two-bit residual budget, we tailor the quantizer to the resulting prediction errors. At this low precision, concentrated coordinates make quantization sensitive to outliers (Sanjeet et al., 2026). Hadamard rotations can spread outliers across channels (Ashkboos et al., 2024); transformed LLM weights and activations can also become approximately Gaussian (Malinovskii et al., 2025; Egiazarian et al., 2026). The effect of a fixed block Hadamard transform depends on input geometry and transform width (Sanjeet et al., 2026, Proposition 3.2 and Corollary 3.3). On our prediction residuals, wider transforms can



(c) H_{16} versus H_{64} residual error

Residual	H_{16} variation	H_{64} variation	H_{64} error change ↓
HY K	0.433	0.291	**-7.49%**
HY V	0.299	0.300	+0.03%
MAGI K	0.359	0.295	**-2.96%**
MAGI V	0.280	0.278	-0.02%

Variation: std./mean of subblock squared magnitudes.

Figure 5: (a–b) Median absolute excess kurtosis across layers and two prompts; lower is closer to a unit Gaussian. H_n transforms each non-overlapping group of n residual channels independently. Wider transforms have model-dependent effects beyond H_{64}. (c) Comparison at one shared scale per 64 values. "Variation" is the standard deviation divided by the mean of the squared magnitudes of the four 16-value subblocks; lower values indicate more uniform magnitudes.

instead concentrate correlated structure in a few coefficients. It is therefore unclear how many channels should be transformed together. We test independent transforms over groups of $n = 4$ to 1024 channels, denoted H_n. Figure 5(a–b) measures absolute excess kurtosis (zero for a Gaussian). H_{16} and H_{64} perform most consistently across both models; larger groups help MAGI-1 but give HY-WorldPlay heavier tails, especially for values.

Figure 5(c) compares H_{16} and H_{64} with one scale per 64 residual values. H_{16} transforms four 16-value groups; H_{64} transforms the entire group. H_{64} makes key magnitudes more even across the four groups, reducing key-residual quantization error by 7.49% on HY-WorldPlay and 2.96% on MAGI-1; value-residual error changes by less than 0.03%. At the same bit allocation, H_{64} improves mean video PSNR by 0.69 and 0.25 dB, respectively. We use H_{64} and assign each normalized, transformed residual pair to one of 16 codewords in the fixed, Gaussian-tuned EDEN-2 codebook (Vargaftik et al., 2022). An inverse transform reconstructs the residuals before they are added in Eq. 1.

To compare with QVG while assigning the same number of bits to residual codes and scales, we target 2.125 bits per value. Each prediction residual uses a two-bit code, leaving 0.125 bits per value for scales. For keys, we store one 8-bit scale per 64 residual values; for values, we store one 16-bit scale per 128 residual values. Our ablations show that keys are more sensitive than values to sharing a scale across adjacent blocks, which motivates this allocation. We specify the quantizer in Appendix C.1 and report the full ablations in Appendix H.2.

5 Experimental Methodology

Models, data, and baselines. We evaluate CORAL-KV on two autoregressive video generators. HY-WorldPlay generates 285 frames at 704×1280 resolution using four denoising steps. Due to resource constraints, its evaluations retain only 48 latent frames of context. MAGI-1 generates 192 frames at 720×720 resolution using 16 denoising steps. We evaluate on 355 held-out MovieGen Extended prompts (Polyak et al., 2024) that are disjoint from those used for calibration and model selection.

Baselines are BF16, QVG-1, QVG-4, QVG-Pro, and an AQUA-KV-based dense cross-layer predictor. QVG-1 uses one centroid-fitting stage per cache block and quantizes its residual in 64-value groups; QVG-4 adds three centroid stages with the same residual group size. QVG-Pro uses four stages and 16-value residual groups (Xi et al., 2026). It is our strongest quality baseline; QVG-4 matches reusable CORAL-KV's residual bit allocation and serves as our latency baseline. We evaluate reusable CORAL-KV with offline-calibrated predictors and video-adapted CORAL-KV with block-diagonal components refit per cache block. Appendix D.1 gives model and split details.

Quality evaluation. We measure reconstruction fidelity against matched BF16 generations using RGB PSNR, SSIM (Wang et al., 2004), and VGG-LPIPS (Zhang et al., 2018). We compress each filled cache block as a unit: 7,040 visual tokens on HY-WorldPlay and 12,150 on MAGI-1. To stress-test sustained generation with compressed history, we score only frames after the first com-

Table 1: **CORAL-KV improves fidelity with less total cache memory in matched comparisons against existing baselines on both models.** Shared predictors are counted once at 16 videos. Latency is generation-time overhead relative to BF16; its measurement protocol appears in Appendix G.3.

	Fidelity			Total memory (GB) ↓		Latency
Method	PSNR ↑	SSIM ↑	LPIPS ↓	1 video	16 videos	Overhead ↓
HY-WorldPlay (285 frames)						
BF16	–	–	–	46.71	747.42	0.0%
QVG-1	15.70	0.496	0.513	7.99	127.91	+7.8%
QVG-4	16.66	0.540	0.438	13.36	213.82	+15.3%
QVG-Pro	17.16	0.559	0.408	14.46	231.34	+17.7%
AQUA-KV	17.12	0.560	0.399	12.66	104.03	+71.7%
CORAL-KV (reusable)	16.98	0.551	0.406	**6.90**	**102.90**	+8.7%
CORAL-KV (video-adapted)	**17.38**	**0.568**	**0.392**	9.74	148.26	+29.1%
MAGI-1 (192 frames)						
BF16	–	–	–	6.77	108.29	0.0%
QVG-1	24.13	0.785	0.250	1.07	17.09	+2.6%
QVG-4	26.14	0.830	0.207	1.58	25.20	+10.3%
QVG-Pro	28.70	0.872	0.162	1.73	27.74	+9.6%
AQUA-KV	28.13	0.861	0.159	1.09	**14.26**	+6.3%
CORAL-KV (reusable)	31.09	0.899	0.127	**0.99**	14.84	+6.8%
CORAL-KV (video-adapted)	**32.47**	**0.913**	**0.111**	1.15	17.35	+10.0%

pressed block begins affecting generation. The first affected frames are 29 and 24, respectively (zero-indexed), so fidelity is evaluated on frames 30–284 and 25–191. We assess compression-boundary discontinuities separately with QBD.

We additionally evaluate temporal consistency using quantization-boundary discontinuity (QBD) and six VBench dimensions (Huang et al., 2024). QBD embeds each frame with DINO ViT-B/16 (Caron et al., 2021) and computes the cosine similarity between adjacent frames. It compares the similarity drop around cache-compression boundaries with the average drop at ordinary locations in the same video, then normalizes this ratio by the matched BF16 generation. Lower values indicate smaller compression-induced temporal discontinuities. Appendix E.1 provides the formal definition and validation experiments.

Memory and latency evaluation. For HY-WorldPlay, we report total cache memory for the 285-frame generation. For MAGI-1, we compute the total cache memory used within its context limit. Totals include residuals, scales, assignments, anchors, predictors or codebooks, and video-specific parameters. We measure generation time on 16 matched prompts using identical attention implementations and include any video-specific fitting time. Appendix G.3 provides the complete memory accounting and latency protocol.

6 MAIN RESULTS

CORAL-KV achieves higher fidelity with less cache memory. Table 1 compares reusable CORAL-KV with QVG-1 and QVG-4, and video-adapted CORAL-KV with QVG-Pro. Against QVG-1, reusable CORAL-KV improves PSNR by 1.28 dB on HY-WorldPlay and 6.96 dB on MAGI-1, with 13.7% and 7.5% less cache memory, respectively. At the same residual-bit allocation as QVG-4, it improves PSNR by 0.32 dB and 4.95 dB while using 48.4% and 37.3% less cache memory, respectively, and lowers generation overhead on both models. Increasing QVG from one to four centroid stages raises both PSNR and cache memory, but QVG-4 still trails reusable CORAL-KV in PSNR while using more memory on both models.

Video-adapted CORAL-KV improves all three fidelity metrics over QVG-Pro while using less cache memory on both models, including a 3.77 dB gain with 33.7% less memory on MAGI-1 and 32.7% less memory on HY-WorldPlay. Across the QVG-4 and QVG-Pro comparisons, paired 95% bootstrap intervals for PSNR, SSIM, and LPIPS gains exclude zero on both models. The video-adapted option pays for its higher fidelity with online block fitting; a direct matched comparison finds higher generation overhead than reusable CORAL-KV on both models (Appendix G.3). We also compare CORAL-KV with DensityKV, a recent video-cache token-selection method, and find that CORAL-KV significantly improves PSNR by 3.27–5.11 dB at matched compressed-history memory (Appendix F).

Table 2: **CORAL-KV improves boundary stability with less cache memory and broadly similar VBench scores relative to QVG-Pro on both models.** Cache memory (GB) is the single-video total from Table 1. All results use 355 prompts.

Method	Cache ↓	QBD ↓	VBench score ↑					
			Subject	Background	Imaging	Aesthetic	Motion	Flicker
HY-WorldPlay								
BF16	46.71	1.000	89.153	92.821	69.434	61.860	98.208	96.073
QVG-1	7.99	3.346	84.399	91.032	57.982	59.959	**98.508**	**97.263**
QVG-4	13.36	1.386	88.436	92.395	66.869	61.932	98.328	96.683
QVG-Pro	14.46	1.158	**89.317**	**92.932**	68.040	**61.965**	98.274	96.440
AQUA-KV	12.66	**1.003**	88.748	92.557	69.513	61.737	98.128	96.010
CORAL-KV (reusable)	**6.90**	1.022	88.555	92.558	**69.604**	61.925	98.105	95.998
CORAL-KV (video-adapted)	9.74	1.026	88.791	92.700	69.166	61.874	98.155	96.085
MAGI-1								
BF16	6.77	1.000	97.218	97.302	64.205	64.904	99.433	98.976
QVG-1	1.07	3.499	96.538	96.820	63.488	63.975	99.298	98.849
QVG-4	1.58	3.194	96.828	97.012	63.637	64.445	99.361	98.893
QVG-Pro	1.73	2.527	97.203	97.227	64.188	65.030	99.405	98.964
AQUA-KV	1.09	3.537	97.092	97.205	64.085	64.778	99.411	98.950
CORAL-KV (reusable)	**0.99**	2.150	97.329	97.297	**64.647**	65.240	99.434	98.996
CORAL-KV (video-adapted)	1.15	**1.328**	**97.351**	**97.340**	64.606	**65.301**	**99.439**	**99.004**

At 16 videos, shared predictors give reusable CORAL-KV lower total memory than QVG-1 and QVG-4 on both models. HY-WorldPlay's full-history total is 6.90 GB versus 13.36 GB for QVG-4 (Table 1), pointing toward longer-context video generation. The AQUA-KV comparison illustrates the value of compact prediction: on wide HY-WorldPlay caches, video-adapted CORAL-KV uses 23.1% less full-history memory and cuts generation overhead from 71.7% to 29.1%. On MAGI-1, our AQUA-KV baseline uses the published 1,024-value Hadamard groups (Shutova et al., 2025) with one FP16 scale per group. Its residual format costs 2.016 rather than CORAL-KV's 2.125 bits per value, making its 16-video total slightly smaller (14.26 versus 14.84 GB). Reusable CORAL-KV uses less memory for one video and improves PSNR by 2.96 dB. Appendix G.2 gives the memory-sharing details.

Lower boundary discontinuity with similar VBench scores. Table 2 shows that both configurations reduce quantization-boundary discontinuity relative to QVG-Pro on both models under all 12 predeclared definitions (Appendix E.1); video-adapted CORAL-KV nearly halves it on MAGI-1 (1.328 versus 2.527). They use 32.7–52.3% less single-video cache memory (Table 1) while achieving broadly similar VBench scores. The largest observed VBench gain is HY-WorldPlay imaging quality, up 1.1–1.6 points. Appendix Figures 14–18 show further examples of the first compression boundary on both models.

Although QVG has higher VBench motion-smoothness and flicker scores on HY-WorldPlay, unusually smooth frames can produce such scores. To check for suppressed temporal variation, we measure the mean absolute pixel difference between adjacent post-quantization frames on 32 prompts after grayscale downsampling, normalized to each video's paired BF16 generation. QVG shows 13% less frame-to-frame change than BF16, whereas CORAL-KV nearly matches the BF16 frame-to-frame change. Thus QVG's higher scores partly reflect suppressed variation (Appendix E.2); Appendix Figures 10–13 show related examples of smoothing and detail loss.

7 CONCLUSION

We introduce CORAL-KV, a KV-cache compression framework for autoregressive video generation that predicts cache states across layers and quantizes the remaining residual. A predictor bank calibrated once offline can be shared across videos for an efficient default, while optional per-block adaptation trades additional memory and fitting time for higher fidelity. On HY-WorldPlay and MAGI-1, both configurations move the measured fidelity–memory frontier beyond matched clustering baselines, showing that better reconstruction need not require a larger cache. Reusing the offline bank also amortizes its cost across concurrent video caches, so total cache memory grows more slowly with video count. Together, these results point toward high-fidelity video generation with smaller, more scalable KV caches.

AI USE STATEMENT

In this work, we used AI to help refine hypotheses and conceptual framing; provide feedback on experimental methodology; assist with implementation and debugging; organize experimental outputs; and support qualitative and quantitative interpretation of results. We additionally used it for brainstorming, literature discovery, citation verification, figure and table preparation, and drafting and editing parts of the manuscript. We did not use generative AI to generate evaluation datasets, formulate or prove mathematical claims, or independently determine scientific conclusions. The authors reviewed all AI-assisted work, tested code and numerical outputs, checked reported results against evaluation outputs, and verified citations against primary sources. We take responsibility for the final content of this work, including text, claims, code, figures, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies compression of internal state in existing video-generation models and does not introduce human-subject data or train a new generative model. The method inherits the potential biases, misuse risks, and content limitations of the underlying generators; reducing memory use and inference cost does not mitigate those risks and may make such systems easier to use. Our evaluation concerns reconstruction fidelity, memory, and latency rather than the safety or factual reliability of generated content. Users should therefore retain the safeguards and usage policies of the underlying models.

REPRODUCIBILITY STATEMENT

Calibration, ablation, and 355-prompt evaluation splits are disjoint. Every comparison uses matching model revisions, prompts, seeds, and conditioning inputs. We treat prompts as the statistical units and form confidence intervals with paired prompt-level bootstrap resampling. The appendix reports cache-memory accounting and the latency protocol. Code and evaluation data will be released.

REFERENCES

Juan Amboage, Pablo Monteagudo-Lago, Ian Colbert, Giuseppe Franco, and Nicholas Fraser. Optimal post-training quantization scales and where to find them. *arXiv preprint arXiv:2606.10890*, 2026. URL https://arxiv.org/abs/2606.10890.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.52202/079017-3180. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b5b939436789f76f08b9d0da5e81af7c-Abstract-Conference.html.

William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan-Kelley. Reducing transformer key-value cache size with cross-layer attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.52202/079017-2758. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/9e23d020c18e4c40d81c6a0fc7a46f68-Abstract-Conference.html.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://openaccess.thecvf.com/content/ICCV2021/html/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.html.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Hung-Yueh Chiang, Yash Akhauri, Xilai Dai, Huiqiang Jiang, Yucheng Li, Luis Ceze, Kai-Chiang Wu, and Mohamed S. Abdelfattah. xKV: Cross-layer KV-cache compression via aligned singular vector extraction. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2503.18893.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.52202/075280-0196. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/0df38cd13520747e1e64e5b123a78ef8-Abstract-Conference.html.

Boyuan Chen, Diego Martí Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2407.01392.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. SkyReels-V2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025. URL https://arxiv.org/abs/2504.13074.

Yukang Chen, Luozhou Wang, Wei Huang, Shuai Yang, Bohan Zhang, Yicheng Xiao, Ruihang Chu, Weian Mao, Qixin Hu, Shaoteng Liu, Yuyang Zhao, Huizi Mao, Ying-Cong Chen, Enze Xie, Xiaojuan Qi, and Song Han. LongLive-2.0: An NVFP4 parallel infrastructure for long video generation. *arXiv preprint arXiv:2605.18739*, 2026. URL https://arxiv.org/abs/2605.18739.

Ian Colbert, Eashan Dash, Pablo Monteagudo-Lago, Juan Amboage, Srinidhi N, Giuseppe Franco, Nicholas J. Fraser, and Arun Ramachandran. Signed symmetric quantization for few-bit integers. *arXiv preprint arXiv:2607.08779*, 2026. URL https://arxiv.org/abs/2607.08779.

Anna Cordoba, Adam Puente Tercero, Nerea Angulo Hijo, Mar Linares Tercero, Julia Barrientos, Ainhoa Miranda, and Jesus Olivera. DepthWeave-KV: Token-adaptive cross-layer residual factorization for long-context KV cache compression. *arXiv preprint arXiv:2607.06523*, 2026. URL https://arxiv.org/abs/2607.06523.

DeepSeek-AI et al. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024. URL https://arxiv.org/abs/2405.04434.

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

Vage Egiazarian, Roberto Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Bridging the gap between promise and performance for microscaling FP4 quantization. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b87bb4f6346d727b265088235e5bc389-Abstract-Conference.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.17323.

Xiaomeng Fu, Jia Li, Yiming Hu, Yong Wang, Hayden Kwok-Hay So, Jiao Dai, Xiangxiang Chu, and Jizhong Han. Towards memory-efficient autoregressive video generation via instance-specific parametric absorption. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2607.00712.

Google DeepMind. Veo: A text-to-video generation system. Technical report, Google DeepMind, 2025. URL https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Cyrus Wu, Wei Li, Xuchen Song, Yang Liu, Eric Li, and Yahui Zhou. Matrix-Game 2.0: An open-source, real-time, and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025. URL https://arxiv.org/abs/2508.13009.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. ZipCache: Accurate and efficient KV cache quantization with salient token identification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.52202/079017-2181. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7e57131fdeb815764434b65162c88895-Abstract-Conference.html.

Yicong Hong, Yiqun Mei, Chongjian Ge, Yiran Xu, Yang Zhou, Sai Bi, Yannick Hold-Geoffroy, Mike Roberts, Matthew Fisher, Eli Shechtman, Kalyan Sunkavalli, Feng Liu, Zhengqi Li, and Hao Tan. RELIC: Interactive video world model with long-horizon memory. *arXiv preprint arXiv:2512.04040*, 2025. URL https://arxiv.org/abs/2512.04040.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2401.18079.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. URL https://arxiv.org/abs/2506.08009.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://arxiv.org/abs/2311.17982.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b. doi: 10.1109/TPAMI.2025.3633890. URL https://arxiv.org/abs/2411.13503.

Shiyu Ji, Yixuan Wang, Yijun Liu, Qingfu Zhu, and Wanxiang Che. EchoKV: Efficient KV cache compression via similarity-based reconstruction. *arXiv preprint arXiv:2603.22910*, 2026a. URL https://arxiv.org/abs/2603.22910.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-KV: Hybrid KV cache compression for efficient autoregressive video diffusion models. *arXiv preprint arXiv:2605.09681*, 2026b. URL https://arxiv.org/abs/2605.09681.

Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. GEAR: An efficient error reduction framework for KV cache compression in LLM inference. In *Proceedings of the 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, volume 262 of *Proceedings of Machine Learning Research*, pp. 305–321. PMLR, 2024. URL https://proceedings.mlr.press/v262/kang24a.html.

Suyoung Kim, Sunghyun Wee, Hyeonjin Kim, Kyomin Hwang, Hyunho Lee, and Nojun Kwak. ReSpinQuant: Efficient layer-wise LLM quantization via subspace residual rotation approximation. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2604.11080.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. URL https://arxiv.org/abs/2412.03603.

Rahul Krishnan and Volker Schulz. A JoLT for the KV cache: Near-lossless KV cache compression via joint lagrangian allocation of tucker ranks and a rotated residual for LLMs. *arXiv preprint arXiv:2607.12550*, 2026. URL https://arxiv.org/abs/2607.12550.

Kunyang Li, Mubarak Shah, and Yuzhang Shang. PackCache: A training-free acceleration method for unified autoregressive video generation via compact KV-cache. *arXiv preprint arXiv:2601.04359*, 2026. URL https://arxiv.org/abs/2601.04359.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial attention: O(n log n) sparse attention with energy decay for long video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2506.19852.

Hongzhan Lin, Zhiqi Bai, Xinmiao Zhang, Sen Yang, Xiang Li, Siran Yang, Yunlong Xu, Jiaheng Liu, Yongchi Zhao, Jiamang Wang, Yuchi Xu, Wenbo Su, and Bo Zheng. Reconstructing KV caches with cross-layer fusion for enhanced transformers. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2512.03870.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. MiniCache: KV cache compression in depth dimension for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. doi: 10.52202/079017-4443. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/fd0705710bf01b88a60a3d479ea341d9-Abstract-Conference.html.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2405.16406.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *International Conference on Machine Learning (ICML)*, 2024b. URL https://arxiv.org/abs/2402.02750.

Vladimir Malinovskii, Andrei Panferov, Ivan Ilin, Han Guo, Peter Richtárik, and Dan Alistarh. HIGGS: Pushing the limits of large language model quantization via the linearity theorem. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 10857–10886, 2025. doi: 10.18653/v1/2025.naacl-long.543. URL https://aclanthology.org/2025.naacl-long.543/.

Meituan LongCat Team et al. LongCat-Video technical report. *arXiv preprint arXiv:2510.22200*, 2025. URL https://arxiv.org/abs/2510.22200.

Kepan Nan, Wangbo Zhao, Penghao Zhou, Jun Li, Zhenheng Yang, Jian Yang, and Ying Tai. Accelerating autoregressive video diffusion via history-guided cache and residual correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43740–43750, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Nan_Accelerating_Autoregressive_Video_Diffusion_via_History-Guided_Cache_and_Residual_Correction_CVPR_2026_paper.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://arxiv.org/abs/2304.07193.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. URL https://arxiv.org/abs/2410.13720.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.01801.

Sand.ai et al. MAGI-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025. URL https://arxiv.org/abs/2505.13211.

Sai Sanjeet, Ian Colbert, Pablo Monteagudo-Lago, Giuseppe Franco, Yaman Umuroglu, and Nicholas J. Fraser. Pushing the limits of block rotations in post-training quantization. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2601.22347.

Semyon Savkin, Eitan Porat, Or Ordentlich, and Yury Polyanskiy. NestQuant: Nested lattice quantization for matrix products and LLMs. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.09720.

Jinliang Shen, Lianghao Su, Zheming Li, Kang He, ZiLiang Lai, Yanbing Jiang, and Chengru Song. HeadCast: Casting attention heads for efficient autoregressive video generation. *arXiv preprint arXiv:2607.20125*, 2026. URL https://arxiv.org/abs/2607.20125.

Alina Shutova, Vladimir Malinovskii, Vage Egiazarian, Denis Kuznedelev, Denis Mazur, Nikita Surkov, Ivan Ermakov, and Dan Alistarh. Cache me if you must: Adaptive key-value quantization for large language models. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55451–55473. PMLR, 2025. URL https://proceedings.mlr.press/v267/shutova25a.html.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. You only cache once: Decoder-decoder architectures for language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.52202/079017-0235. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/0df38cd13520747e1e64e5b123a78ef8-Abstract-Conference.html.

Keda Tao, Haoxuan You, Yang Sui, Can Qin, and Huan Wang. Plug-and-play 1.x-bit KV cache quantization for video large language models. *arXiv preprint arXiv:2503.16257*, 2025. URL https://arxiv.org/abs/2503.16257.

Team Wan et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

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)*, 2024. URL https://arxiv.org/abs/2402.04396.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001. doi: 10.1023/A:1017501703105. URL https://www.mit.edu/~dimitrib/PTseng/papers/archive/bcr_jota.pdf.

Tuna Tuncer, Felix Becker, and Thomas Pfeil. Quantized keys steal attention: Bias correction for KV-cache compression in video diffusion. *arXiv preprint arXiv:2605.26266*, 2026. URL https://arxiv.org/abs/2605.26266. Variants accepted to the ICML 2026 workshops SCALE and F2S.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric and challenges. *arXiv preprint arXiv:1812.01717*, 2018. URL https://arxiv.org/abs/1812.01717.

Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben-Itzhak, and Michael Mitzenmacher. EDEN: Communication-efficient and robust distributed mean estimation for federated learning. In *International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2108.08842.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Thaddäus Wiedemer, Yuxuan Li, Paul Vicol, Shixiang Shane Gu, Nick Matarese, Kevin Swersky, Been Kim, Priyank Jaini, and Robert Geirhos. Video models are zero-shot learners and reasoners. *arXiv preprint arXiv:2509.20328*, 2025. URL https://arxiv.org/abs/2509.20328.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. HunyuanVideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025. URL https://arxiv.org/abs/2511.18870.

You Wu, Ziheng Chen, Yizhen Zhang, Haoyi Wu, Chengting Yu, Yuchi Xu, Wenbo Su, Bo Zheng, and Kewei Tu. YOCO++: Enhancing YOCO with KV residual connections for efficient LLM inference. *arXiv preprint arXiv:2604.13556*, 2026. URL https://arxiv.org/abs/2604.13556.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse VideoGen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.01776.

Haocheng Xi, Shuo Yang, Yilong Zhao, Muyang Li, Han Cai, Xingyang Li, Yujun Lin, Zhuoyang Zhang, Jintao Zhang, Xiuyu Li, Zhiying Xu, Jun Wu, Chenfeng Xu, Ion Stoica, Song Han, and Kurt Keutzer. Quant VideoGen: Auto-regressive long video generation via 2-bit KV-cache quantization. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.02958.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.17453.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. WORLDMEM: Long-term consistent world simulation with memory. *arXiv preprint arXiv:2504.12369*, 2025. URL https://arxiv.org/abs/2504.12369.

Haoqi Yang, Yao Yao, Zuchao Li, Baoyuan Qi, Guoming Liu, and Hai Zhao. XQuant: Achieving ultra-low bit KV cache quantization with cross-layer compression. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9785–9800, 2025a. doi: 10.18653/v1/2025.emnlp-main.494. URL https://aclanthology.org/2025.emnlp-main.494/.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025b. URL https://arxiv.org/abs/2509.22622.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025c. URL https://arxiv.org/abs/2505.18875.

Hidir Yesiltepe, Jiazhen Hu, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, Hoda Eldardiry, and Pinar Yanardag. VideoMLA: Low-rank latent KV cache for minute-scale autoregressive video diffusion. *arXiv preprint arXiv:2605.30351*, 2026. URL https://arxiv.org/abs/2605.30351.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2412.07772.

Amir Zandieh, Majid Daliri, and Insu Han. QJL: 1-bit quantized JL transform for KV cache quantization with zero overhead. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 25805–25813, 2025. doi: 10.1609/aaai.v39i24.34773. URL https://ojs.aaai.org/index.php/AAAI/article/view/34773.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online vector quantization with near-optimal distortion rate. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2504.19874.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. URL https://arxiv.org/abs/1801.03924.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2306.14048.

Wenqu Zhao, Xuemin Chi, Xin Zhang, Guoqing Ma, Baorun Li, Jianjie Fang, Peizhi Tang, Chen Gao, and Wei Wu. DensityKV: Density-guided KV cache compression for long video generation. *arXiv preprint arXiv:2608.27922*, 2026. URL https://arxiv.org/abs/2608.27922.

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 (MLSys)*, 2024. URL https://arxiv.org/abs/2310.19102.

A LIMITATIONS

We evaluate two autoregressive video models and one held-out prompt source; gains vary with cache geometry, and we do not formally explain why linear cross-layer prediction works or establish that it transfers to other models, prompt distributions, or longer horizons. Memory results cover stored cache state rather than whole-process peak GPU memory. The 16-video totals use shared-bank accounting validated with up to four retained videos (Appendix G.2), not measured concurrent throughput; generation latency was measured on MI355X. CORAL-KV retains every historical token position, leaving integration with token selection open. We interpret no-reference VBench scores alongside paired BF16 fidelity and motion because smoothness scores alone can reward motion suppression.

B EXPANDED RELATED WORK

Autoregressive video generation. Causal and diffusion-forcing formulations generate video in temporal blocks and retain prior context (Chen et al., 2024; Yin et al., 2025; Huang et al., 2025a). Long-form and interactive systems include MAGI-1, LongLive, LongLive-2.0, SkyReels-V2, LongCat-Video, and Matrix-Game 2.0 (Sand.ai et al., 2025; Yang et al., 2025b; Chen et al., 2026; 2025; Meituan LongCat Team et al., 2025; He et al., 2025); related backbones and explicit-memory systems include Wan, HunyuanVideo, HunyuanVideo 1.5, and WORLDMEM (Team Wan et al., 2025; Kong et al., 2024; Wu et al., 2025; Xiao et al., 2025).

Video-cache compression and attention reduction. QVG directly quantizes autoregressive video K/V with online codebooks, and Quantized Keys Steal Attention studies the associated attention bias (Xi et al., 2026; Tuncer et al., 2026). Forcing-KV and DensityKV select retained cache content; PackCache compacts tokens; FastAR combines temporal compression with sparse attention; VideoMLA and HeadCast alter the stored attention state; and ISPA absorbs history into instance-specific parameters (Ji et al., 2026b; Zhao et al., 2026; Li et al., 2026; Samuel et al., 2026; Yesiltepe et al., 2026; Shen et al., 2026; Fu et al., 2026). ARCache instead reuses denoising features (Nan et al., 2026). Sparse VideoGen, Sparse VideoGen2, and Radial Attention reduce attention computation and are complementary to value compression (Xi et al., 2025; Yang et al., 2025c; Li et al., 2025). Low-bit video-LLM cache quantization provides another video-domain analogue (Tao et al., 2025).

LLM KV compression and cross-layer structure. KIVI, KVQuant, ZipCache, and GEAR quantize or correct LLM cache state, while TurboQuant and QJL develop online projection/vector-quantization schemes (Liu et al., 2024b; Hooper et al., 2024; He et al., 2024; Kang et al., 2024; Zandieh et al., 2026; 2025). AQUA-KV predicts the next layer and quantizes its residual (Shutova et al., 2025). Cross-Layer Attention, MiniCache, xKV, XQuant, DepthWeave-KV, FusedKV, and EchoKV instead share, align, factorize, or reconstruct state across depth (Brandon et al., 2024; Liu et al., 2024a; Chang et al., 2026; Yang et al., 2025a; Cordoba et al., 2026; Lin et al., 2026; Ji et al., 2026a). JoLT jointly allocates Tucker ranks and codes a rotated residual (Krishnan & Schulz, 2026). Architectural approaches such as YOCO, YOCO++, and multi-head latent attention reduce the cache by design (Sun et al., 2024; Wu et al., 2026; DeepSeek-AI et al., 2024); token-selection approaches such as H2O and StreamingLLM are orthogonal (Zhang et al., 2023; Xiao et al., 2024).

Quantization and residual coding. GPTQ is an influential post-training LLM quantization method; QuIP, QuIP#, QuaRot, SpinQuant, and ReSpinQuant use incoherence or learned rotations to control outliers (Frantar et al., 2023; Chee et al., 2023; Tseng et al., 2024; Ashkboos et al., 2024; Liu et al., 2025; Kim et al., 2026). EDEN supplies the Gaussian-tuned quantization grid used by HIGGS; later work studies scale selection, small-block rotations, signed grids, nested lattices, and rate–distortion coding (Vargaftik et al., 2022; Malinovskii et al., 2025; Amboage et al., 2026; Sanjeet et al., 2026; Colbert et al., 2026; Savkin et al., 2025; Domb et al., 2026). Atom studies low-bit serving-oriented quantization (Zhao et al., 2024).

Evaluation protocols. Our prompt suite comes from MovieGen (Polyak et al., 2024). We report SSIM and LPIPS fidelity (Wang et al., 2004; Zhang et al., 2018), use DINO features for boundary

analysis (Caron et al., 2021), and use dimensions from VBench (Huang et al., 2024). VBench++, FVD, and DINOv2 provide related evaluation and representation-learning context (Huang et al., 2025b; Unterthiner et al., 2018; Oquab et al., 2024).

C Method and implementation details

C.1 Full quantization and reconstruction specification

We index the cache by patch tokens and define N contiguous positions as a *cache block*; in our implementations, a cache block contains one or more complete latent frames. Completed blocks are encoded once and retained in compressed form, while entries that are not yet eligible remain at native precision. For $K_\ell, V_\ell \in \mathbb{R}^{N\times C}$, layer zero is encoded and reconstructed as

$$Z_0^K = \mathrm{Enc}_{\mathrm{A}}(K_0), \qquad Z_0^V = \mathrm{Enc}_{\mathrm{A}}(V_0), \tag{2}$$

$$\widehat{K}_0 = \mathrm{Dec}_{\mathrm{A}}(Z_0^K), \qquad \widehat{V}_0 = \mathrm{Dec}_{\mathrm{A}}(Z_0^V). \tag{3}$$

For every $\ell > 0$, the layer-specific predictors produce

$$\widetilde{K}_\ell = P_\ell^K(\widehat{K}_{\ell-1}), \qquad E_\ell^K = K_\ell - \widetilde{K}_\ell, \tag{4}$$

$$\widetilde{V}_\ell = P_\ell^V([\widehat{K}_\ell, \widehat{V}_{\ell-1}]), \qquad E_\ell^V = V_\ell - \widetilde{V}_\ell, \tag{5}$$

where brackets concatenate channels. The residual codes and reconstructions are

$$Z_\ell^K = \mathrm{Enc}_{\mathrm{R}}(E_\ell^K), \qquad Z_\ell^V = \mathrm{Enc}_{\mathrm{R}}(E_\ell^V), \tag{6}$$

$$\widehat{K}_\ell = \widetilde{K}_\ell + \mathrm{Dec}_{\mathrm{R}}(Z_\ell^K), \qquad \widehat{V}_\ell = \widetilde{V}_\ell + \mathrm{Dec}_{\mathrm{R}}(Z_\ell^V). \tag{7}$$

Reconstruction proceeds in layer order, with keys preceding values because P_ℓ^V depends on $\widehat{K}_\ell$. During encoding, reconstructed rather than target tensors feed the next predictor, so encoder and decoder inputs match. The cache retains only Z_0^K, Z_0^V and $\{Z_\ell^K, Z_\ell^V\}_{\ell=1}^{L-1}$ with their scales and decoder metadata.

For completeness, the anchor's per-channel affine scalar quantizer maps a value x in channel c to

$$z = \mathrm{clip}\left(\mathrm{round}\left(\frac{x-\beta_c}{\alpha_c}\right), 0, 2^b - 1\right), \qquad \widehat{x} = \alpha_c z + \beta_c, \tag{8}$$

with FP16 scale and offset. At low bitwidth, independent scalar quantization can be dominated by outliers or energy concentrated in a few coordinates. Randomized orthogonal rotations can reduce coordinate outliers (Vargaftik et al., 2022; Tseng et al., 2024). Hadamard-rotated LLM weights and activations have been observed to approach Gaussian distributions (Malinovskii et al., 2025; Egiazarian et al., 2026); related rotations have also been used for KV caches (Ashkboos et al., 2024). A fixed Hadamard transform has no such input-independent guarantee, so we select its width empirically.

We distinguish scale-group size g from transform size $h \mid g$. For $u \in \mathbb{R}^g$, store $s = \|u\|_2/\sqrt{g/h}$, partition u into g/h blocks $u_i \in \mathbb{R}^h$, and compute

$$y_i = H_h(u_i/s), \qquad H_h H_h^\top = hI. \tag{9}$$

Partitioning transformed coordinates into p-vectors y_j, assignment to the nearest entry in a fixed codebook $\mathcal{C} = \{c_1, \ldots, c_M\}$ is

$$z_j = \arg\min_{m\in\{1,\ldots,M\}} \|y_j - c_m\|_2^2, \tag{10}$$

and decoding concatenates the selected vectors and applies

$$\widehat{u}_i = \frac{s}{h} H_h^\top \widehat{y}_i. \tag{11}$$

With a b_s-bit stored scale, the storage cost is $\log_2(M)/p + b_s/g$ bits/value before metadata. We use the published EDEN codebook with $p = 2$, $M = 16$, and $h = 64$. The selected key format uses $(b_s, g) = (8, 64)$ and the value format uses $(16, 128)$, giving 2.125 bits/value for either keys or values. The uniform FP16-per-64 control uses 2.25 bits/value. Unlike analyses that prepend random signs, we use the same fixed block Hadamard matrix throughout the channel dimension. We therefore evaluate its distributional effect and select the transform width empirically in Appendix H.2.

Model geometry and offline fitting. The effective predictor geometry is $(L, H_{\mathrm{KV}}, d, C) = (30, 24, 256, 6144)$ for HY-WorldPlay and $(34, 8, 128, 1024)$ for MAGI-1. Offline calibration pools all 384,000 and 1,024,000 cached token positions, respectively, from the 32 calibration rollouts. We fit each layer's block-diagonal term D_ℓ and low-rank factors A_ℓ, B_ℓ with three rounds of alternating headwise ridge and reduced-rank regression updates, using regularization 10^{-3} and seed zero. Normal equations are accumulated in FP64 and solved by Cholesky, while the inputs and stored predictor factors are BF16. On one MI355X, the complete fitting procedure took 24.8 minutes for HY-WorldPlay and 29.6 minutes for MAGI-1, including data loading, sequential reconstruction, auditing, and checkpointing but excluding generation of the calibration rollouts. The stored predictors occupy 502.5 and 69.3 MB and are included in reported cache memory.

We use three fitting rounds for all reported predictors. In a HY-WorldPlay screen with disjoint calibration and evaluation cache dumps, six and ten rounds reduced held-out KV relative MSE by 4.97% and 7.55% relative to three rounds. The ten-round predictor improved PSNR by 0.10 dB on 32 selection-set videos (95% CI $[-0.21, 0.41]$); its LPIPS and BF16-relative temporal differences also spanned zero. We therefore retained three rounds for the reported method.

Video-specific adaptation. The video-adapted configuration retains the reusable rank-128 low-rank factors A_ℓ and B_ℓ. For each eligible cache block, it fits the block-diagonal term D_ℓ and the intercept by independent per-head ridge solves over every token in the block, again with regularization 10^{-3}. It then fits one least-squares scale and bias per output channel. The solves use FP32 and the returned block, bias, and affine tensors use the native BF16 cache dtype. These parameters are included in memory while their block remains in the active attention history.

Stored residual representation. Each 64-value residual group stores 32 four-bit EDEN-2 indices, because each index reconstructs a pair of transformed values. Key groups store one FP8 norm per 64 values, while value groups store one FP16 norm per 128 values. The key and value residuals therefore each require 2.125 bits/value before alignment and container metadata. Layer zero uses stored four-bit affine codes with one FP16 scale and offset per channel. Reported memory is obtained from the allocated tensors, not from this bit-count calculation. The fixed 16-entry EDEN-2 codebook is shared; its bytes are included once when it is stored rather than embedded as a compile-time constant. We use the published $d = 2, n = 16$ codebook without fitting centroids to our data; its ordered reconstruction points are listed in Table 3. The video-adapted configuration and the uniform-scale ablation instead use one FP16 scale per 64 residual values, for a 2.25-bpv residual format.

C.2 Efficient inference implementation

A direct implementation of Eqs. 4–7 would decode the stored values, form two predictions per layer, add the residuals, and rearrange the result on every cache read. Offline calibration removes predictor fitting; the following optimizations reduce the remaining conversion overhead without changing the representation or reconstruction rule.

Stored representation and factorized prediction. Stored cache blocks contain only anchor or residual codes, group scales, and decoding metadata. Reported memory includes one shared copy of the predictors and codebook; reconstructed K/V and workspaces are transient. The block-diagonal term uses batched per-head contractions and the low-rank factors use thin matrix multiplications, never materializing $D_\ell + A_\ell B_\ell$. The implementation reuses prepared casts and eligible precomputed projections, accepts strided head blocks directly, and accumulates components in place. Synchronized profiles give dense-to-block-DPLR predictor-application ratios of $2.64\times$ on HY-WorldPlay and $0.76\times$ on MAGI-1. We therefore select the factorized form for memory efficiency rather than assuming universal speedup, and use separate unprofiled runs in both execution orders for end-to-end latency claims.

Fused compressed-cache conversion. Where model shapes permit, GPU kernels combine the transform, codebook assignment, encoding, decoding, residual addition, and attention-layout operations. The decoder combines the prediction with the decoded H_{64}/EDEN residual in place and emits the attention-facing layout without retaining an expanded cache block, avoiding intermediate allocations and redundant casts, transposes, and copies. Entries that will not be read again

Table 3: Ordered two-dimensional EDEN-2 reconstruction points used by the residual quantizer. Values are shown to six decimal places.

Index	y_1	y_2	Index	y_1	y_2
0	-0.899663	-1.636042	8	0.172082	-0.940172
1	-0.961183	1.599957	9	-1.759970	-0.624423
2	-1.882026	0.678778	10	-0.899381	0.322678
3	0.363008	-1.966787	11	0.839489	-0.301704
4	-0.681407	-0.576819	12	1.531495	1.294204
5	0.727001	0.618686	13	-0.001178	0.000221
6	0.335942	1.837119	14	1.427453	-1.207889
7	1.859930	0.036669	15	-0.161239	0.878751

Table 4: Evaluation configurations. Attention implementations are held fixed across methods within each model.

Model	Resolution	Frames	Denoising steps	First compression-dependent frame	Attention implementation
HY-WorldPlay	704×1280	285	4	29	AITER
MAGI-1	720×720	192	16	24	native GQA

are skipped. HY-WorldPlay uses approved fused assignment and reconstruction kernels. MAGI-1 uses fused codebook assignment while retaining conservative transform and reconstruction dispatch where broader fusion does not pass the numerical checks. Both implement the same stored representation and reconstruction rule.

Matched attention implementation and numerical checks. All methods within a model use the same attention implementation: AITER for HY-WorldPlay and the native grouped-query attention implementation for MAGI-1. HY-WorldPlay is checked at the reconstructed-tensor and final-BF16-output levels; MAGI-1 additionally uses a paired decoded-video check. Only implementations passing these checks enter the latency study, so differences do not arise from a relaxed algorithm or attention implementation.

D Experimental protocol

D.1 Evaluation configurations

Table 5: Disjoint prompt splits and their permitted uses.

Split	Prompts	Permitted use
Calibration	32	Fit offline predictors and quantizer parameters
Selection	32	Select configurations and report ablations
Evaluation	355	Held-out comparisons and confidence intervals

The split membership is immutable and prompt-length stratified. A comparison is admitted only when model revision, prompt index, seed, conditioning image, generation geometry, and scorer configuration match. HY-WorldPlay methods also share the same action sequence. Offline predictors are fitted only on the calibration split; neither selection nor evaluation cache tensors are used to update them. Configuration choices are fixed after selection-set analysis and before examining the 355-prompt evaluation aggregates.

Calibration prompt identities. The 32 calibration prompts are zero-based indices in the 1,003-prompt MovieGen Extended source file, in calibration order: 490, 342, 4, 145, 796, 598, 240, 789, 744, 718, 782, 632, 812, 128, 338, 925, 757, 807, 199, 917, 406, 653, 968, 161, 591, 554, 941, 601, 573, 952, 940, and 967. The source prompt file has SHA-256 `50293c2717c8f2df5773154d4680c74e636ecb08eb4fe13cdce7357d0ab9d121`.

Layerwise cross-layer-predictability probe. Figure 2(a,b) evaluates full affine ridge predictors on unquantized KV states, using 4,096 evenly spaced token positions per cache dump and a ridge rate of 10^{-3}. Lines average three rollout-held-out evaluations and shading spans those evaluations. For HY-WorldPlay, the key and preceding-value curves fit each held-out rollout from the other 15 rollouts; the combined-input value curve evaluates a fixed full-affine map calibrated on eight disjoint rollouts. For MAGI-1, each of the first three held-out rollouts is predicted by maps fit on the other 15 rollouts. We report $1 - \operatorname{mean}_c \operatorname{Var}_t(\widehat{y} - y)_c / \operatorname{mean}_c \operatorname{Var}_t(y)_c$ at each predicted layer.

KV-compression transition qualitative selection. Figure 2(c) uses HY-WorldPlay prompt 8 from the selection set. Among the 32 matched BF16/QVG-1/reusable-CORAL triples, we ranked the increase in the CORAL-minus-QVG full-frame RGB PSNR gap from frame 28 to the first quantization-affected frame 29, then checked print-scale legibility. Prompt 8 ranked first: the gap changes from -0.19 dB at frame 28 to $+10.53$ dB at frame 29.

E Additional metric and statistical details

E.1 Quantization-boundary discontinuity

For decoded frame x_t, let h_t be its normalized DINO ViT-B/16 embedding and define consecutive-frame consistency for $t = 1, \ldots, T-1$ as

$$s_t = h_t^\top h_{t-1}. \tag{12}$$

Let m be the median consistency within the video. For any eligible decoded frame index u, define the width-w local transition window $\mathcal{W}_w(u) = \{u, \ldots, u+w\}$, clipped at the end of the video, and the corresponding local dip

$$d_w(u) = m - \min_{t \in \mathcal{W}_w(u)} s_t. \tag{13}$$

The boundary set $\mathcal{B}$ follows the cache-quantization schedule and VAE temporal decoding. The interior set $\mathcal{I}_r$ excludes video ends and all locations within radius r of every scheduled chunk boundary. The primary definition uses $w = 2$, corresponding to three transitions, and $r = 3$. We define

$$\mathrm{QBD}(x) = \frac{|\mathcal{B}|^{-1} \sum_{b \in \mathcal{B}} d_2(b)}{|\mathcal{I}_3|^{-1} \sum_{t \in \mathcal{I}_3} d_2(t)}. \tag{14}$$

A value near one means scheduled boundaries are no less consistent than ordinary interior locations; larger values indicate boundary-localized discontinuities. The BF16-normalized diagnostic is $\mathrm{QBD}(x_{\mathrm{compressed}}) / \mathrm{QBD}(x_{\mathrm{BF16}})$ for paired videos under the same boundary schedule. Both CORAL-KV configurations have lower BF16-normalized QBD than QVG-Pro on both models across all tested window widths $w \in \{1, 2, 3\}$ and exclusion radii $r \in \{2, 3, 4, 5\}$.

E.2 VBench and BF16-relative dynamics

VBench-6 diagnostics. We complement reference-based fidelity with a compression-focused subset of six official VBench dimensions (Huang et al., 2024): subject consistency, background consistency, imaging quality, aesthetic quality, motion smoothness, and temporal flickering. We call this subset VBench-6 and report all six coordinates separately as absolute scores; it is not a new aggregate benchmark. Subject and background consistency retain the clean first frame as a global reference, while the other dimensions operate on the contiguous post-quantization window. For HY-WorldPlay, full-reference fidelity uses frames 30–284, while QBD and VBench-6 use frames 29–284. For MAGI-1, VBench-6 uses frames 24–95 so that it remains comparable to the conventional 96-frame setting; full-reference fidelity uses frames 25–191 and QBD uses frames 24–191. Since no-reference motion smoothness and flicker can reward suppressed motion, we also compare temporal dynamics with BF16 before attributing a gain to improved stability.

BF16-relative temporal dynamics. To distinguish genuine stability from motion suppression, we area-downsample the post-quantization window to 64×36 grayscale frames and measure its mean absolute inter-frame difference and spatial total variation. We divide both quantities by the value from the paired BF16 video; ratios near one preserve the reference motion magnitude and spatial

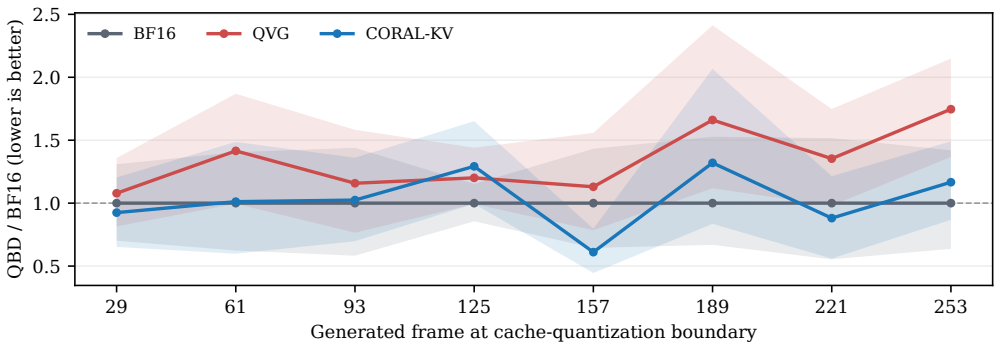


Figure 6: **Boundary stability over long-horizon generation.** BF16-normalized QBD at all eight cache-quantization boundaries on the 32-prompt HY-WorldPlay selection set. Lines are prompt means and bands are 95% bootstrap intervals. Lower is better.

variation. We also measure the absolute difference between the compressed and BF16 frame-to-frame residuals, normalized by the BF16 inter-frame difference, for which zero is ideal. These quantities are diagnostics rather than replacement quality metrics.

E.3 Per-boundary QBD

Aggregate QBD can hide whether discontinuities worsen as generation proceeds. Figure 6 therefore resolves all eight scheduled HY-WorldPlay boundaries on the 32-prompt selection set. Across the complete 285-frame rollout, mean QBD is $1.035\times$ BF16 for CORAL-KV and $1.321\times$ for QVG-4; the paired QVG-minus-CORAL-KV difference in raw QBD is 0.472 (95% CI $[0.242, 0.702]$), with improvements on 26 of 32 prompts. At the final boundary, the corresponding ratios are $1.166\times$ and $1.747\times$. QVG's fitted QBD slope is positive at 0.149 per boundary ($[0.052, 0.253]$), whereas CORAL-KV has a smaller slope of 0.053 whose interval includes zero ($[-0.053, 0.161]$).

F Comparison with token selection

Token selection reduces memory by discarding historical positions rather than compressing their keys and values. We port the released DensityKV selection rule (Zhao et al., 2026) to both video generators and also compare with a control that keeps the most recent historical BF16 tokens. All methods use the same local window, first-frame tokens, generation settings, and paired BF16 references. For each CORAL-KV variant, we cap each selector's K/V and metadata at the same compressed-history memory. On HY-WorldPlay, this active-window budget excludes the common 5.19-GB BF16 suffix; Table 1 instead reports full-history memory.

We use DensityKV's released per-head selection rule based on post-RoPE key density. On HY-WorldPlay, it processes each completed four-frame block. MAGI-1's 12,150-token blocks exceed the matched bank capacity, so we seed the bank with uniformly spaced positions from the first block and process later blocks in chronological 512-token groups. When MAGI-1 rewrites a block, we refresh K/V for retained positions without changing their density metadata. These changes accommodate the models' update schedules; the selection rule is unchanged.

We score HY-WorldPlay frames 30–284 on 16 paired MovieGen prompts and MAGI-1 frames 25–191 on 64 paired prompts, averaging frame PSNR within each prompt before averaging prompts. At matched memory, CORAL-KV exceeds DensityKV by 3.27–5.11 dB and the recent-token control in every row (Table 6). All four paired 95% prompt-bootstrap intervals for the DensityKV gains exclude zero (smallest lower bound 2.76 dB); MAGI-1 SSIM and LPIPS agree. Thus, in these settings, compressing all historical positions preserves more BF16 fidelity than retaining selected BF16 positions. A disjoint eight-prompt HY-WorldPlay follow-up gives the same PSNR ordering. The HY-WorldPlay comparison is exploratory because BF16 outputs vary across processes; each method is compared with its own paired BF16 reference. These tests measure fidelity at matched cache memory, not perceptual quality on DensityKV's original backbones.

Table 6: **CORAL-KV has higher BF16-relative PSNR than both token-selection controls at matched cache memory.** Mean RGB PSNR (dB; higher is better) on 16 HY-WorldPlay and 64 MAGI-1 paired prompts. Memory is the retained compressed-history budget; each control's K/V and any selection metadata fit within it. These cohorts are separate from the 355-prompt evaluation in Table 1.

Model (prompts)	Variant	Memory (GB)	CORAL-KV	DensityKV	Recent BF16
HY-WorldPlay (16)	Reusable	4.05	16.47	11.76	11.52
HY-WorldPlay (16)	Video-adapted	4.52	16.78	11.67	11.75
MAGI-1 (64)	Reusable	0.99	31.45	28.18	27.31
MAGI-1 (64)	Video-adapted	1.15	32.97	28.67	27.38

G MEMORY ACCOUNTING AND IMPLEMENTATION CONTROLS

For method m, reported compressed-history memory is

$$M_m = M_{\text{codes}} + M_{\text{scales}} + M_{\text{metadata}} + M_{\text{pred}} + M_{\text{online}}, \tag{15}$$

where the cache-block terms cover all nine completed HY-WorldPlay blocks or MAGI-1's retained attention window, and M_{pred} contains one complete set of offline predictors, and M_{online} contains block-specific parameters required for future reconstruction. QVG codebooks are included for every encoded block that needs them. We do not divide reusable predictor memory by the number of prompts or videos in the experiment. HY-WorldPlay totals use the checkpoint after nine complete blocks, so no incomplete native-precision suffix is present. The fidelity evaluation instead retains 48 latent frames, as described in Section 5. Reconstructed tensors, attention outputs, and conversion workspaces are temporary and excluded; these analytical cache-memory totals are not peak process-memory measurements. Direct measurements with up to four simultaneously retained videos are reported below. MAGI-1 retains at most four preceding 24-frame chunks (96 previous frames); older blocks and their associated video-specific parameters are excluded for every method because no future query can attend them. For video-adapted CORAL-KV, the current block-diagonal parameters belong to and are stored with the newest compressed block and are counted only as part of that block.

G.1 FULL-HISTORY CACHE SCALING

Table 1 uses a full-history HY-WorldPlay cache that retains every completed 7,040-token block. The 285-frame generation contains nine such blocks (72 latent frames). Each table entry is the total stored KV state immediately after the listed blocks have been compressed, so no incomplete BF16 block is present. QVG retains its codebooks and assignments for every block; CORAL-KV retains the encoded blocks and includes its 0.502-GB predictor bank once. The per-block byte counts are taken from the packed-cache measurements before the 48-latent-frame context begins replacing older blocks. The measured fidelity results use that retained context.

Table 7: **Reusable CORAL-KV reduces the memory cost of retaining full video history.** Each block contains eight latent frames, or 7,040 visual tokens; nine blocks cover the 285-frame HY-WorldPlay video. The totals include QVG codebooks or the reusable predictor bank.

	Total memory (GB)		
Method	1 block	5 blocks	9 blocks
QVG-1	**0.89**	4.44	7.99
QVG-4	1.48	7.42	13.36
QVG-Pro	1.61	8.03	14.46
CORAL-KV (reusable)	1.21	**4.06**	**6.90**
CORAL-KV (video-adapted)	1.53	5.63	9.74

G.2 MEMORY SHARING ACROSS CONCURRENT VIDEOS

Table 1 includes one complete copy of the reusable predictor bank in the one-video total. In concurrent serving, those predictors are shared while every video retains an independent compressed cache.

For N simultaneous videos, the corresponding compressed-cache memory is therefore $B + NC$, where B is the shared predictor memory and C is the per-video compressed-cache memory. QVG instead fits and retains video-specific centroids, so its compressed-cache memory scales as NQ. The $N = 1$ and $N = 16$ columns of Table 1 report $B + C$ and $B + 16C$, respectively, using nine blocks per HY-WorldPlay video.

G.3 Matched latency definition and results

We measure latency overhead in a dedicated matched experiment with identical prompts, seeds, hardware allocation, and attention implementations. For method m, we report its additional generation time over the local BF16 reference,

$$O_m = \frac{\sum_{p \in \mathcal{P}} T_m(p)}{\sum_{p \in \mathcal{P}} T_{\mathrm{BF16}}(p)} - 1, \tag{16}$$

where $\mathcal{P}$ is the matched prompt set. We average O_m over two execution orders and report it as a percentage. Timing includes online fitting and excludes compilation, warm-up, and video export. Table 1 reports overheads from separate matched experiments. In the direct QVG-Pro comparison, reusable CORAL-KV adds 10.7% versus 17.7% on HY-WorldPlay and 6.8% versus 9.6% on MAGI-1. Both execution orders favor CORAL-KV, although two runs do not establish a statistically significant speedup. A separate matched comparison confirms that video adaptation increases overhead relative to reusable CORAL-KV on both models. We do not compare overheads across separate experiments.

H Ablations and robustness

H.1 Configuration selection

We select the adapter structure and learning regime, residual transform and codebook, and layer-zero representation once on the 32-prompt selection set. It is disjoint from calibration and held-out evaluation; all comparisons use matched prompts, seeds, and conditioning inputs and score complete generated videos using reconstructed caches rather than cache-space regression alone.

For the no-predictor comparison in Figure 4, the rank-128 offline predictor raises PSNR from 14.14 to 17.55 dB on HY-WorldPlay and from 25.19 to 31.99 dB on MAGI-1. The paired gains are 3.41 dB (95% CI $[2.76, 4.12]$) and 6.81 dB ($[6.25, 7.39]$), with 32/32 wins. All four paired intervals for block-DPLR against block-diagonal-only and low-rank-only predictors exclude zero. Across ranks 16 to 128, offline block-DPLR rises from 17.08 to 17.55 dB and from 28.84 to 31.99 dB; rank 256 leaves both means unchanged to two decimals. Low-rank only remains within 14.39–14.61 and 25.37–25.53 dB. Video-specific block-diagonal fitting continues to rank 256, reaching 17.89 and 33.85 dB, at higher memory and fitting cost.

We also tested which predictor component to refit online. In an exploratory eight-video HY-WorldPlay diagnostic, refitting the rank-128 low-rank component instead of the block-diagonal component changed PSNR by only $+0.10$ dB and slightly worsened LPIPS (0.329 versus 0.327). In a separate rank-64 timing profile, the global low-rank update took $2.6\times$ as long to fit as the block-diagonal update, whose independent per-head solves can be parallelized. These diagnostics predate the final residual-quantization configuration and motivate the selected adaptation path rather than serving as headline evidence.

The rank sweep intentionally varies predictor memory. To isolate structure at a fixed budget, we additionally fit the largest standalone low-rank predictors not exceeding block-DPLR r128. All other components and all 32 paired prompts are fixed. Table 8 shows that the standalone controls use slightly fewer bytes, yet block-DPLR has a positive paired PSNR interval on both models. SSIM and MSE also favor block-DPLR on both; HY LPIPS and QBD intervals cross zero, while all five MAGI-1 metrics favor block-DPLR with intervals excluding zero. Thus the gain is not explained by a larger predictor-memory budget.

Layer-zero anchor. Holding the selected predictor and residual quantization fixed, BF16 increases cache memory from 4.85 to 5.18 bits per BF16 cache value on HY-WorldPlay and from 2.47 to

Table 8: **Equal-memory structure control on the 32-prompt selection set.** Deltas are standalone low-rank minus block-DPLR r128; brackets are paired 95% CIs.

Model	LR rank	LR/DPLR MB	ΔPSNR (dB)	ΔSSIM
WorldPlay	281	501.4/502.5	$-0.519\,[-0.867, -0.179]$	$-0.0125\,[-0.0247, -0.0005]$
MAGI-1	204	69.1/69.3	$-2.778\,[-3.240, -2.344]$	$-0.0328\,[-0.0435, -0.0235]$

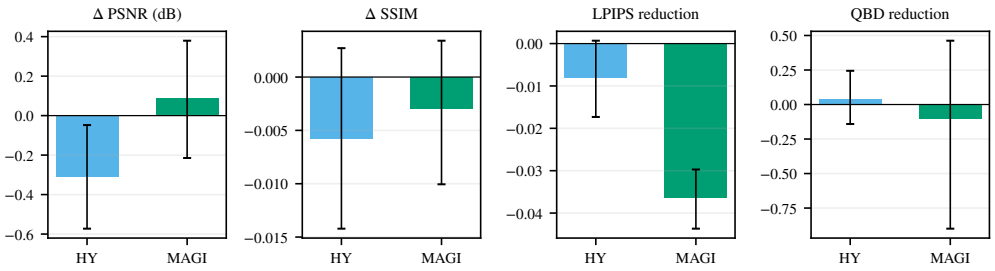


Figure 7: **Layer-zero anchor ablation on the 32-prompt selection set.** Bars show the paired effect of replacing the INT4 anchor with BF16; positive values favor BF16. Error bars are paired 95% bootstrap confidence intervals over 32 prompts.

2.82 on MAGI-1. It changes PSNR by -0.31 dB on HY-WorldPlay (95% CI $[-0.57, -0.05]$) and $+0.09$ dB on MAGI-1 (CI $[-0.21, 0.38]$), while worsening MAGI-1 LPIPS by 0.036 (CI $[0.030, 0.044]$). Thus BF16 adds memory without a quality advantage; Figure 7 reports the paired ablation.

H.2 Residual transform and codebook

We next hold the predictor and layer-zero anchor fixed and vary the residual quantization. To measure generated-video quality, we hold the 64-value scale group fixed and cross transform sizes $h \in \{16, 64\}$ with a symmetric scalar grid and the EDEN-2 grid. This controlled 2×2 comparison uses the same bit allocation and separates the effect of broader Hadamard mixing from vector rather than scalar assignment. Figure 8(a) shows the 16 EDEN-2 reconstruction points over representative H_{64}-transformed residual pairs, and Figure 8(b,c) reports the resulting video fidelity.

We test sharing each FP16 scale across 128 rather than 64 values. This reduces total cache memory by only 2.1% on HY-WorldPlay and 4.9% on MAGI-1, while lowering PSNR by 0.23 and 0.82 dB and increasing LPIPS by 0.0080 and 0.0094, respectively. We therefore reject uniform FP16 groups of 128. We next evaluate three predeclared 2.125-bpv assignments. Relative to the reversed allocation (K FP16/128, V FP8/64), the selected K FP8/64, V FP16/128 format improves PSNR by 0.21/0.67 dB and reduces LPIPS by 0.0080/0.0063 on HY-WorldPlay/MAGI-1. Relative to uniform FP16/128, it gains 0.08/0.85 dB and reduces LPIPS by 0.0041/0.0087. These comparisons motivated the key/value-specific scale allocation, which we then held fixed for held-out evaluation. Uniform FP16 groups of 64 are used by the video-adapted configuration and ablations.

H.3 Calibration scaling

Each row evaluates predictors fitted on the indicated number of calibration videos against the same 32-prompt selection set. MAGI-1 improves steadily, whereas HY-WorldPlay is non-monotonic and peaks at 16 videos. We use 32 calibration videos for both models.

I Additional qualitative results

I.1 Long-horizon robustness

Our main MAGI-1 evaluation generates 192 decoded frames for all 355 held-out prompts. This covers the conventional 96-frame setting and a continuation of equal length without changing prompts,

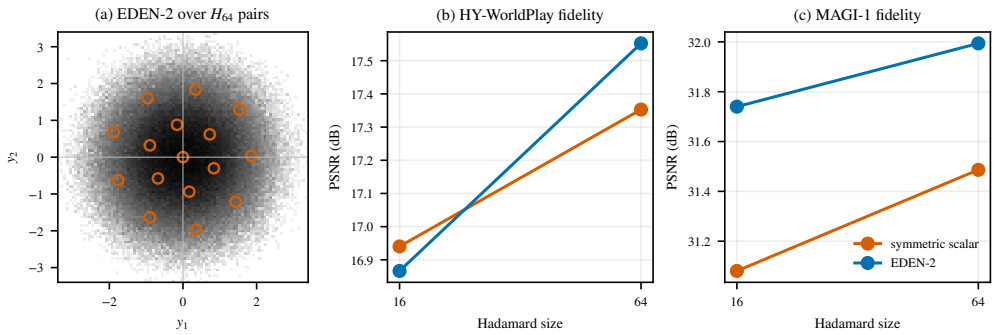


Figure 8: **EDEN-2 codebook and quantizer selection.** (a) The 16 EDEN-2 reconstruction points over representative H_{64}-transformed residual pairs from MAGI-1. (b–c) On the 32-prompt selection set, crossing H_{16}/H_{64} with scalar/EDEN-2 assignment at a fixed 64-value scale group shows the highest PSNR for H_{64}/EDEN-2 on both models.

Table 9: **Calibration-set scaling on the 32-prompt selection set.** Values are PSNR (dB) for rank-128 block-DPLR predictors. MAGI-1 uses the 96-frame setting and is not directly comparable with the primary 192-frame results.

Calibration videos	2	8	16	32
HY-WorldPlay	17.58	18.02	**18.59**	17.94
MAGI-1	37.22	37.89	38.06	**38.39**

seeds, conditioning inputs, or any compression setting. We report the standard post-quantization window (frames 24–95), early continuation (96–143), and late continuation (144–191). The selected reusable and video-adapted configurations are compared with QVG-Pro at the prompt level using 95% bootstrap confidence intervals. Before the full evaluation, we verified the extended rollout on a fixed subset of the selection set without using method outputs for selection.

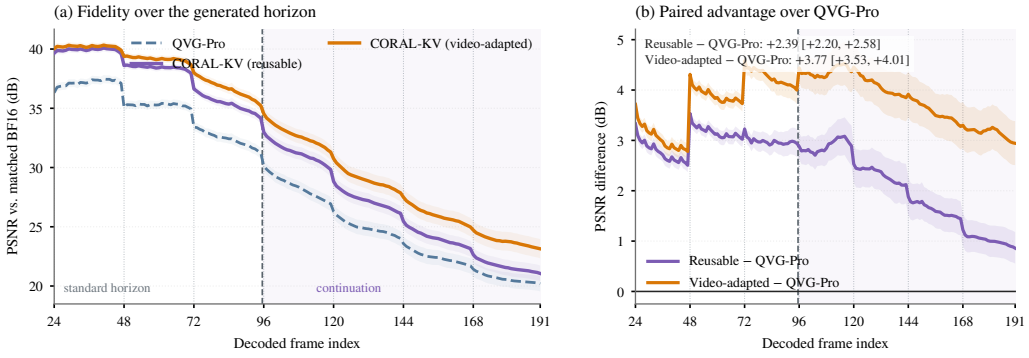


Figure 9: **MAGI-1 fidelity over 192 decoded frames.** Across all 355 held-out prompts, (a) prompt-mean PSNR and 95% bootstrap bands are measured against matched BF16 generation for QVG-Pro and the two reported CORAL-KV configurations; (b) shows paired CORAL-minus-QVG-Pro PSNR differences and intervals. Dotted lines mark cache boundaries; the dashed line begins the continuation beyond the standard 96-frame horizon. Curves are unsmoothed.

Figure 9 shows that the predictor advantage persists beyond the standard horizon for both reported CORAL-KV configurations. On frames 24–95, reusable and video-adapted CORAL-KV improve over QVG-Pro by 2.97 dB (95% CI $[2.85, 3.10]$; 351/355 wins) and 3.74 dB ($[3.59, 3.88]$; 354/355 wins). The corresponding gains are 2.61 and 4.23 dB on frames 96–143, and remain 1.32 dB ($[1.02, 1.63]$; 250/355 wins) and 3.36 dB ($[2.95, 3.78]$; 297/355 wins) in the late frames 144–191.

SSIM and LPIPS agree in every window; in the late window, reusable/video-adapted CORAL-KV improve SSIM by 0.0355/0.0675 and reduce LPIPS by 0.0345/0.0638. The frames 25–191 aggregate reproduces the main-table gains of 2.39 and 3.77 dB. Thus the selected reusable result is not an artifact of the older FP16-per-64 control, although its advantage narrows late in the rollout. A predeclared 32-prompt BF16 usability check found no blank output or rollout collapse through frame 191, so the late-window comparison is not explained by catastrophic failure of the uncompressed generator.

I.2 ADDITIONAL RECONSTRUCTION EXAMPLES

Figures 10–13 show illustrative examples of the quality difference between QVG-Pro and video-adapted CORAL-KV, alongside matched BF16 references.

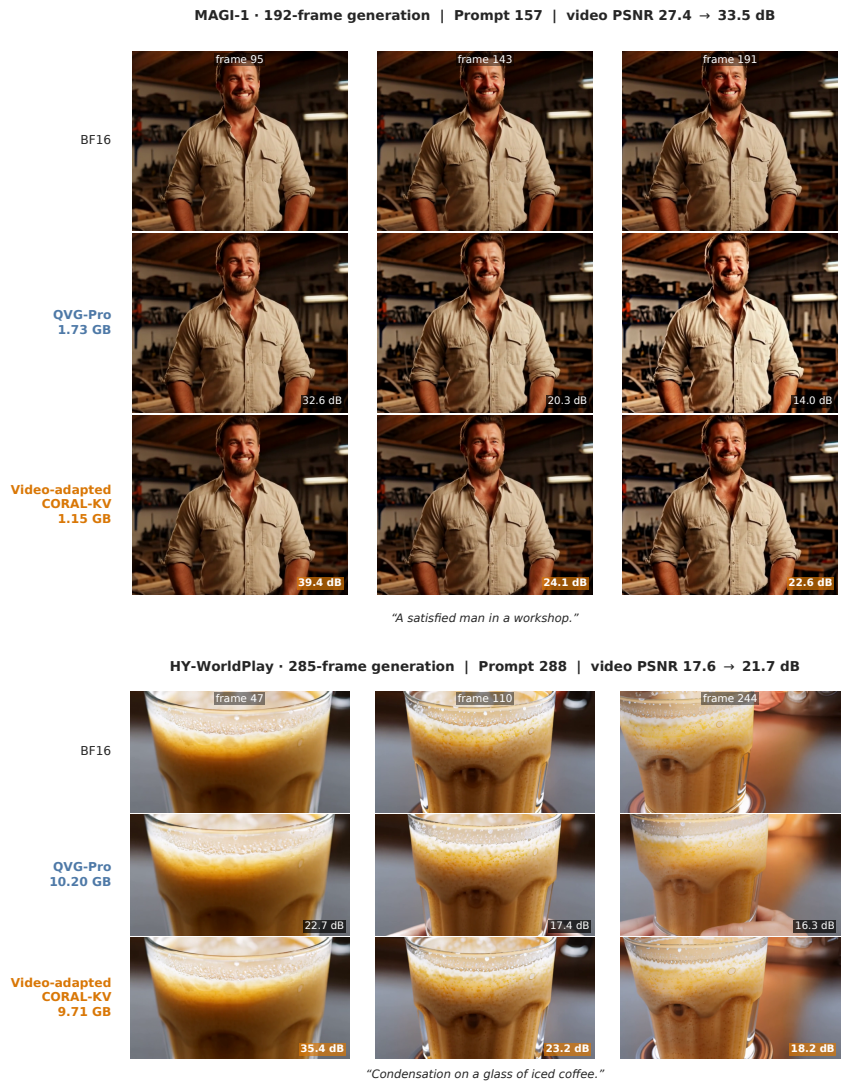


Figure 10: **Long-range reconstruction examples.** MAGI-1 (top; frames 95, 143, and 191) and HY-WorldPlay (bottom; frames 47, 110, and 244). Rows compare BF16, QVG-Pro, and video-adapted CORAL-KV; tile labels report full-frame PSNR against BF16.

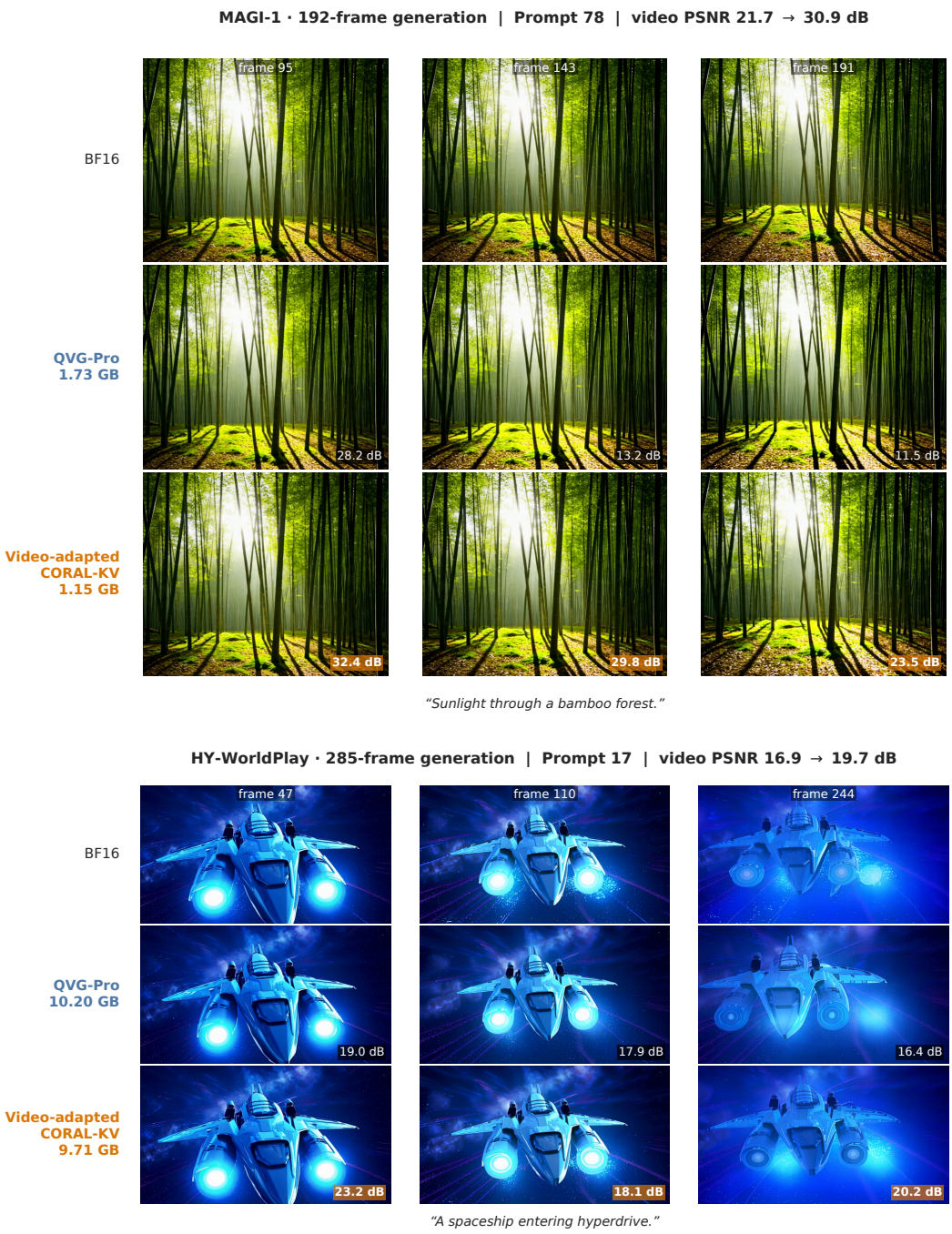


Figure 11: **Long-range reconstruction examples.** MAGI-1 (top; frames 95, 143, and 191) and HY-WorldPlay (bottom; frames 47, 110, and 244). Rows compare BF16, QVG-Pro, and video-adapted CORAL-KV; tile labels report full-frame PSNR against BF16.

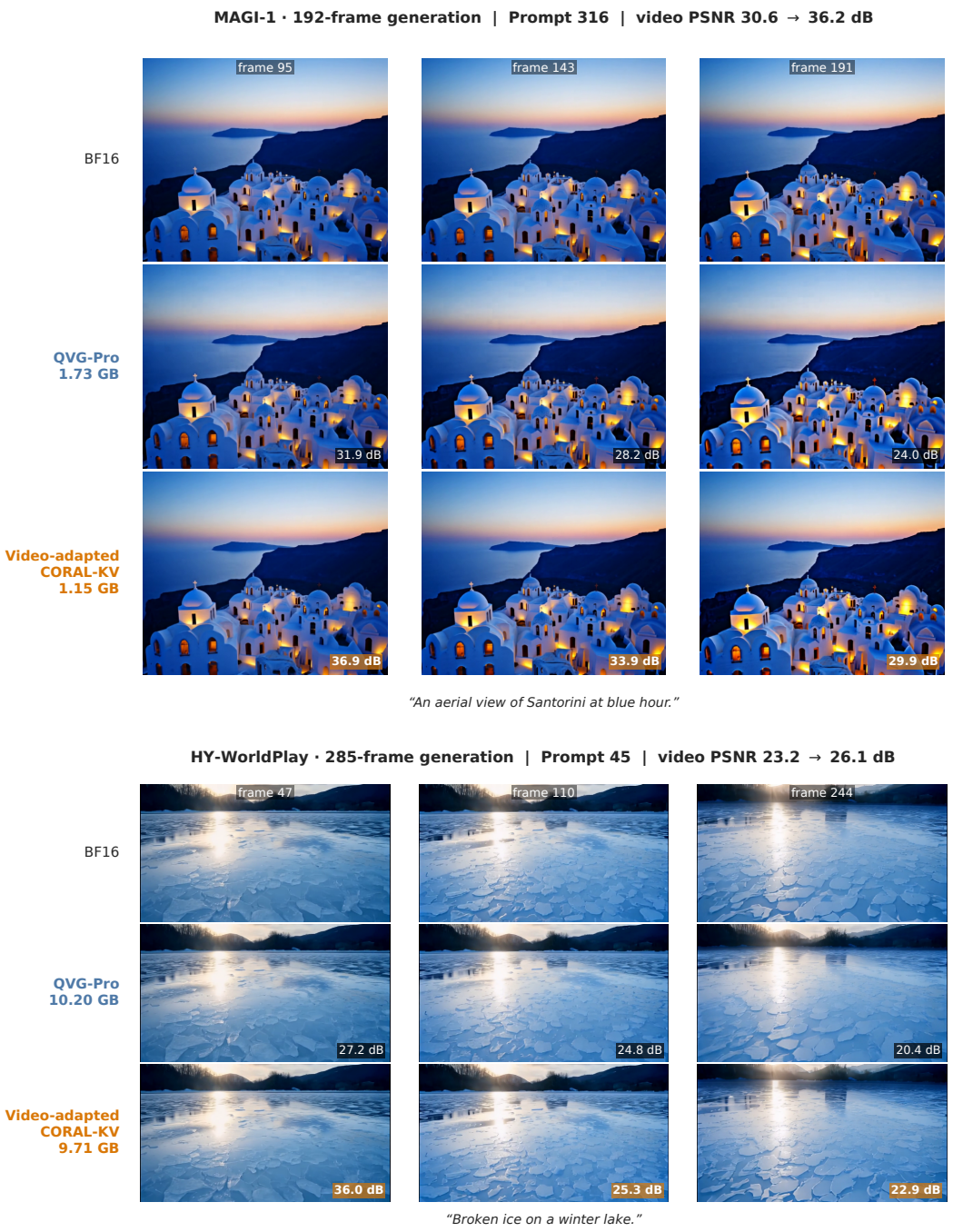


Figure 12: **Long-range reconstruction examples.** MAGI-1 (top; frames 95, 143, and 191) and HY-WorldPlay (bottom; frames 47, 110, and 244). Rows compare BF16, QVG-Pro, and video-adapted CORAL-KV; tile labels report full-frame PSNR against BF16.

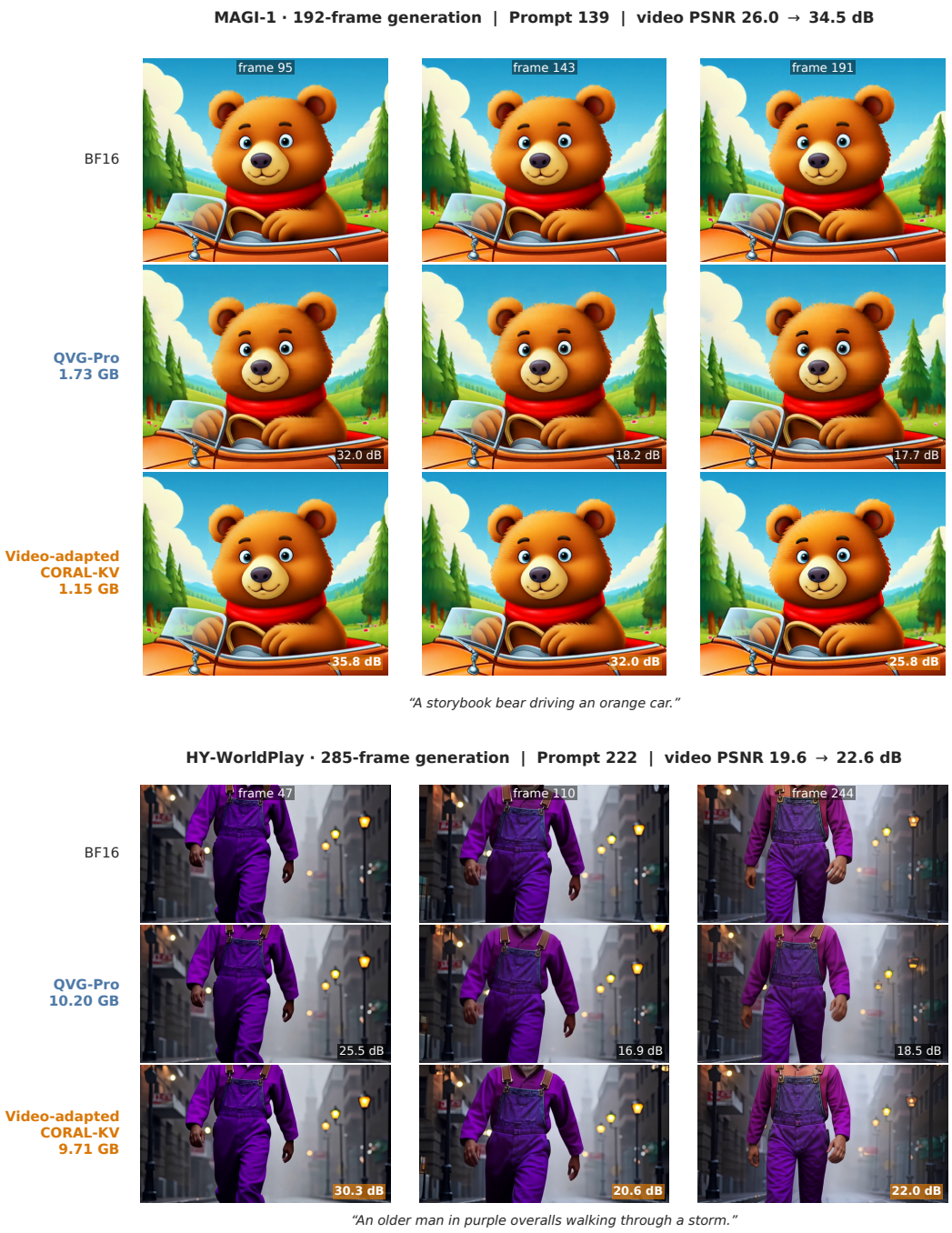


Figure 13: **Long-range reconstruction examples.** MAGI-1 (top; frames 95, 143, and 191) and HY-WorldPlay (bottom; frames 47, 110, and 244). Rows compare BF16, QVG-Pro, and video-adapted CORAL-KV; tile labels report full-frame PSNR against BF16.

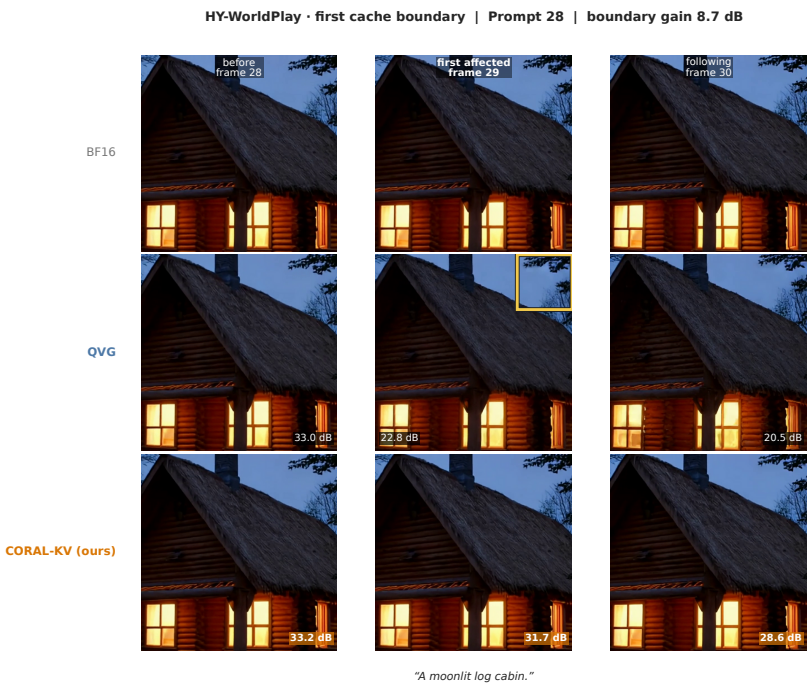


Figure 14: **HY-WorldPlay first-boundary reconstruction.** BF16, QVG, and CORAL-KV at frames 28–30. The yellow box marks QVG's largest local fidelity gap at frame 29; tile labels give PSNR versus BF16.

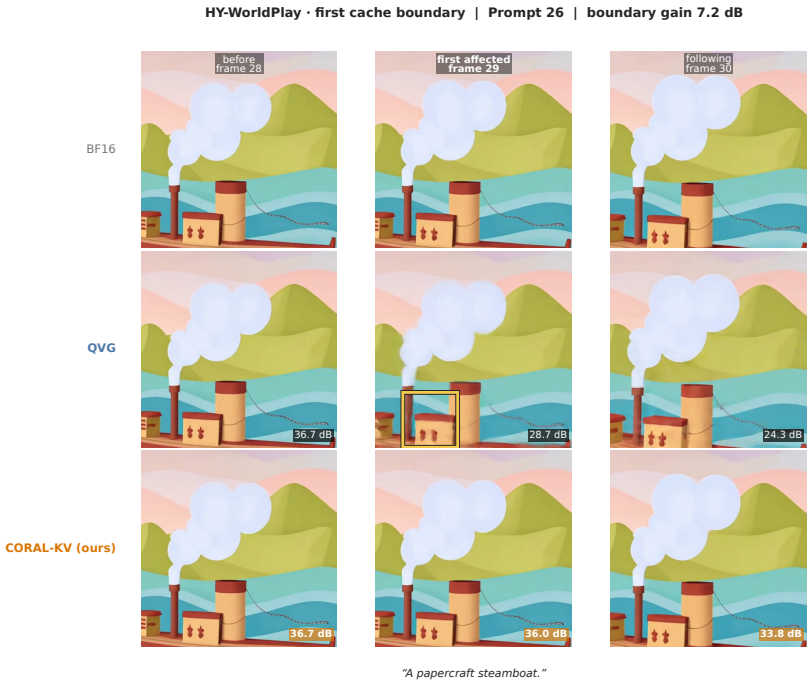


Figure 15: **HY-WorldPlay first-boundary reconstruction.** BF16, QVG, and CORAL-KV at frames 28–30; tile labels give PSNR versus BF16.

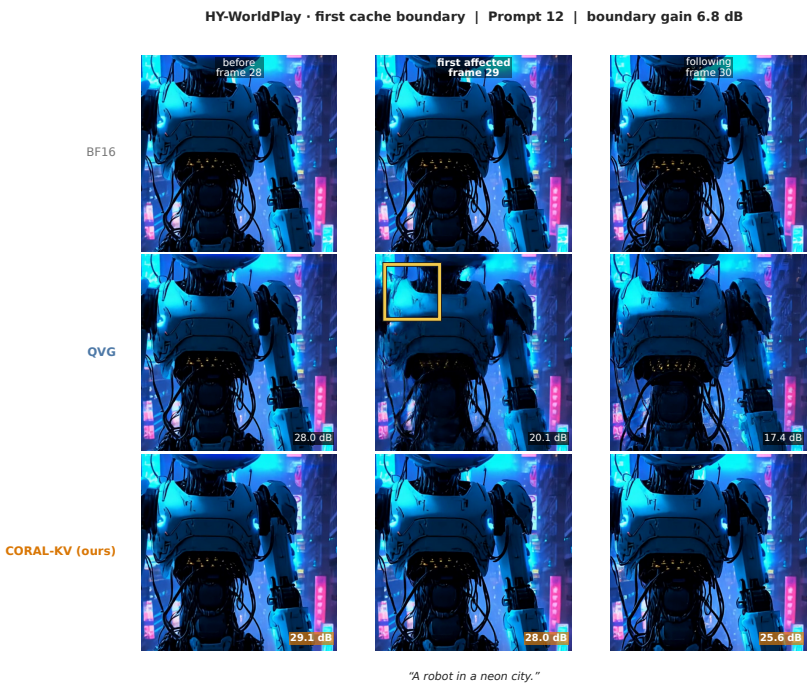


Figure 16: **HY-WorldPlay first-boundary reconstruction.** BF16, QVG, and CORAL-KV at frames 28–30; tile labels give PSNR versus BF16.



Figure 17: **MAGI-1 first-boundary reconstruction.** BF16, QVG, and CORAL-KV at frames 23–25. The yellow box marks QVG's largest local fidelity gap at frame 24; tile labels give PSNR versus BF16.

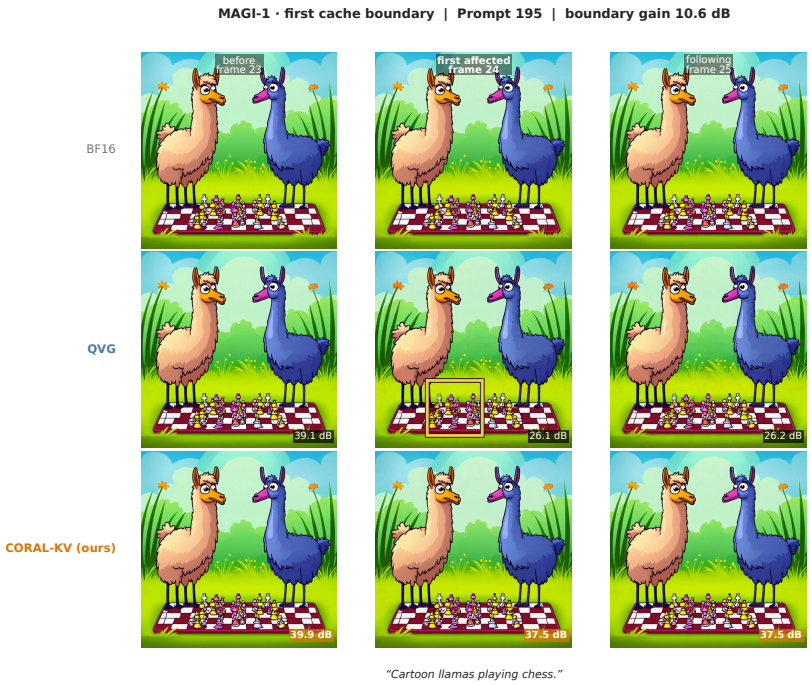


Figure 18: **MAGI-1 first-boundary reconstruction.** BF16, QVG, and CORAL-KV at frames 23–25; tile labels give PSNR versus BF16.