GVCC: Zero-Shot Video Compression via Codebook-Driven Stochastic Rectified Flow

Ziyue Zeng* Xun Su* Haoyuan Liu Bingyu Lu Yui Tatsumi
Hiroshi Watanabe
Graduate School of Fundamental Science and Engineering, Waseda University
Tokyo, Japan
zengziyue@fuji.waseda.jp, suxun_opt@asagi.waseda.jp

Abstract

At ultra-low bitrates, high-fidelity reconstruction requires sampling plausible videos from the posterior rather than regressing to oversmoothed conditional means. We propose Generative Video Codebook Codec (GVCC), a zero-shot framework in which a pretrained video generative model serves directly as the decoder, and the transmitted bitstream specifies its generation trajectory. Modern rectified-flow video models are typically sampled with deterministic ODE solvers, which leave no per-step stochastic channel for transmitting compressed information. GVCC addresses this by converting the deterministic flow sampler into an equivalent marginal-preserving stochastic process, so that information can be transmitted by encoding the per-step stochastic innovations. Unlike images, videos introduce longer temporal dependencies and more diverse conditioning modes. We instantiate GVCC in three practical modes: Text-to-Video (T2V) without a reference frame, autoregressive Image-to-Video (I2V) with tail latent correction, and First-Last-Frame-to-Video (FLF2V) with boundary-sharing Group of Pictures (GOP) chaining. On the seven-sequence UVG dataset, local atom-count sweeps characterize the rate–quality behavior of all three variants. We report full-dataset perceptual and fidelity metrics together with temporal diagnostics, without inferring matched-rate or global RD improvements from these limited local sweeps. Code is available at https://github.com/CCdydy/GVCC.

1 Introduction

Video compression at extremely low bitrates remains a long-standing challenge Lu et al. [2019], Mentzer et al. [2022], Li et al. [2024], Khan et al. [2025]. As bitrate decreases, both conventional hybrid codecs and learned neural codecs tend to suffer from severe detail loss and oversmoothing, a manifestation of the rate–distortion–perception trade-off Blau and Michaeli [2019]. Recent advances in generative modeling have opened a promising direction for perceptual reconstruction. However, most existing generative video compression methods do not make generation the central mechanism of compression. Instead, they adopt a hybrid design in which a conventional codec first produces a compressed representation, and a generative model is then applied as a refinement or restoration module. In such systems, the generative model may improve perceptual quality, but it does not determine the transmitted symbols or the decoder's reconstruction trajectory.

In the image domain, generative compression has evolved from GAN-based learned decoders Goodfellow et al. [2020], Esser et al. [2021], Agustsson et al. [2019], Mentzer et al. [2020] to diffusion-based generative priors Ho et al. [2020], Rombach et al. [2022]. Early diffusion codecs such as DiffC Theis et al. [2022] transmit noisy observations and reconstruct images with a generative prior, while

*Equal contribution.

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

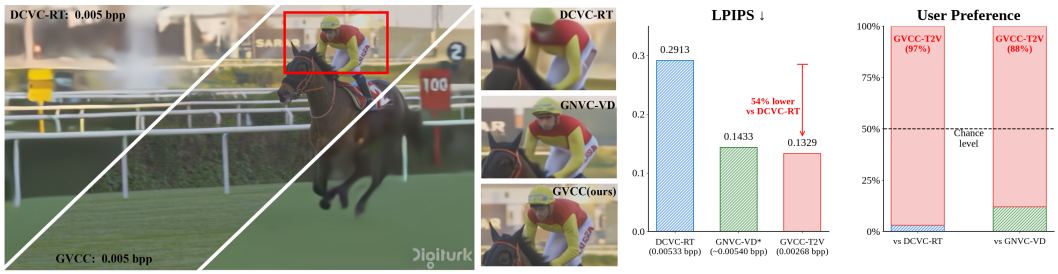

Figure 1: **Qualitative, perceptual, and user-preference comparison.** Left: the *Jockey* example at the displayed sequence-specific operating points. Middle: full-UVG LPIPS-Alex; DCVC-RT (0.2913 at 0.00533 bpp) and GVCC-T2V (0.1329 at 0.00268 bpp) use our unified evaluation, while GNVC-VD* (0.1433 at ∼0.00540 bpp) is read from its published curve. At these displayed points, GVCC-T2V has 54% lower LPIPS than DCVC-RT. Right: the user study (Appendix H) preferred GVCC-T2V to DCVC-RT and GNVC-VD in 97% and 88% of comparisons, respectively.

conditional diffusion decoders further improve rate–perception trade-offs Yang and Mandt [2023], Hoogeboom et al. [2023], Careil et al. [2024]. More recent codecs extend this paradigm to large pretrained image diffusion models Relic et al. [2024], Vonderfecht and Liu [2025], Jia et al. [2026]. At the same time, the generative backbones used by frontier image and video models have shifted from classical diffusion parameterizations toward rectified-flow formulations Liu et al. [2022], Esser et al. [2024], which are now standard in modern video generators. Recent video compression methods have begun to incorporate pretrained video generators Blattmann et al. [2023], Yang et al. [2025], WanTeam et al. [2025] for sequence-level latent refinement Mao et al. [2025], improving temporal consistency. Nevertheless, these approaches still follow a hybrid paradigm: a separate latent codec defines the compressed representation, while the video generator acts as a post-hoc restorer rather than as the codec itself.

This suggests a more direct formulation: rather than adding a generative refinement stage on top of a separately designed codec, we use the pretrained video generative model *as* the codec itself, with the transmitted bitstream directly specifying the decoder's generative trajectory. We refer to this framework as **Generative Video Codebook Codec (GVCC)**.

A natural starting point is frame-wise compression with an image generative codec. However, temporal coherence is a property of the joint distribution across frames rather than of individual marginals, and independent per-frame generation cannot capture it: our preliminary experiments exhibit severe temporal flickering and appearance drift across GOPs. The compression backbone must therefore model the full video joint distribution, motivating the use of video foundation models.

Codebook-driven compression, following DDCM Ohayon et al. [2025], NCS Su and Kasai [2025], and Turbo-DDCM Vaisman et al. [2025], treats the decoder's per-step stochastic innovation as the channel through which information is transmitted. Modern video generators such as Wan 2.1 WanTeam et al. [2025] are built on rectified flow Liu et al. [2022], whose deterministic ODE sampler leaves no such channel once the initial latent is fixed. Crucially, the score-SDE framework Song et al. [2021b] shows that a probability-flow ODE admits a family of reverse-time SDEs with the same continuous-time marginals. Introducing a nonzero diffusion coefficient can therefore be viewed as reallocating stochasticity from the initial latent to per-step innovations, thereby restoring the channel required by codebook-driven compression entirely at inference time, without retraining the generator.

Building on this unified SDE-codebook backbone, we instantiate three conditioning strategies occupying different points in the trade-off between spatial fidelity, temporal coherence, and bitrate. **T2V (Text-to-Video)** uses no reference frame, so the bitrate consists entirely of codebook indices, representing the lowest-side-information regime. **I2V (Image-to-Video)** uses a single reference frame with autoregressive GOP chaining to provide spatial anchoring across segments. **FLF2V (First-Last-Frame-to-Video)** uses both boundary frames of each GOP as dual anchors and constrains generation as interpolation across the segment.

2 Related Work

Generative Video Compression. Neural video codecs, building on learned image compression backbones Ballé et al. [2018], Minnen et al. [2018], achieve strong rate–distortion performance Habibian et al. [2019], Hu et al. [2021], Sheng et al. [2022], Qi et al. [2025], Jia et al. [2025], but at ultra-low bitrates they often yield oversmoothed reconstructions with limited perceptual realism. Recent methods address this issue by incorporating generative priors into the decoding pipeline. GLVC Guo et al. [2025] performs latent-space transform coding with spatio-temporal modeling, GNVC-VD Mao et al. [2025] uses a pretrained video diffusion transformer for sequence-level latent refinement, and GVC Chen et al. [2026] shifts more reconstruction burden to receiver-side generative inference; these methods follow a hybrid paradigm where a separate codec defines the compressed representation and the generative model serves as a refinement module. Free-GVC Ling et al. [2026] departs from this hybrid pattern by encoding directly along the diffusion trajectory via reverse channel coding, sharing the trajectory-based zero-shot philosophy of GVCC; the two differ in compression mechanism and backbone—reverse channel coding on CogVideoX-2B for Free-GVC, versus codebook-driven SDE sampling on rectified-flow video models for GVCC.

Zero-Shot Codebook-Driven Compression. DDCM, Turbo-DDCM and NCS show that a pretrained image diffusion model can be repurposed as a zero-shot image codec by replacing per-step Gaussian noise with reproducible codebook atoms and transmitting only compact indices. However, these methods assume a stochastic reverse process with explicit noise injection at every step. This assumption does not hold for modern rectified-flow video models such as Wan 2.1, whose sampling follows a deterministic ODE Liu et al. [2022]. Currently, Kim et al. [2025] also convert a rectified-flow ODE into an equivalent SDE at inference time, but does so to improve generation diversity and quality for inference scaling. GVCC addresses this mismatch by converting rectified-flow sampling into an equivalent stochastic process at inference time, enabling codebook-driven compression for pretrained video generators without retraining.

3 Method

As shown in Fig.2, GVCC compresses each Group of Pictures (GOP) through a unified pipeline: a pretrained rectified-flow video model is converted into a stochastic process at inference time, and the per-step noise is replaced by reproducible codebook selections whose indices form the transmitted bitstream. All three conditioning variants—T2V,I2V and FLF2V—share the same SDE-codebook backbone and differ only in how the generative model is conditioned. We first review the stochastic-interpolant / flow-matching background and the RF specialization (Sec. 3.1), then describe ODE-to-SDE conversion (Sec. 3.2), score-from-velocity (Sec. 3.3), and codebook-driven discretization (Sec. 3.4), before detailing the three conditioning strategies (Sec. 3.5–3.7).

3.1 Background: stochastic interpolants, flow matching, and rectified flow

Diffusion models Ho et al. [2020], Song et al. [2021a], Karras et al. [2022], flow matching, and rectified flow can all be viewed as learning dynamics along a family of probability paths $\{p_t\}_{0\le t\le 1}$ that interpolate between a data distribution p_0 and a noise distribution p_1 Albergo et al. [2023], Lipman et al. [2023]. A standard construction uses a *stochastic interpolant* between two independent samples $\mathbf{x}_0 \sim p_0$ and $\mathbf{x}_1 \sim p_1$:

$$\mathbf{x}_t = \alpha_t\,\mathbf{x}_0 + \sigma_t\,\mathbf{x}_1, \quad t \in [0,1], \tag{1}$$

where α_t and σ_t are smooth scalars satisfying $\alpha_0{=}1$, $\sigma_0{=}0$, $\alpha_1{=}0$, $\sigma_1{=}1$, together with $\dot\alpha_t < 0$ and $\dot\sigma_t > 0$ so that $\mathbf{x}_t$ bridges p_0 and p_1 Albergo and Vanden-Eijnden [2023], Albergo et al. [2023]. Different choices of (α_t, σ_t) induce different sampling trajectories; rectified flow Liu et al. [2022] corresponds to the *linear* choice

$$\alpha_t = 1 - t, \qquad \sigma_t = t, \tag{2}$$

which yields straight paths in sample space and recovers the usual rectified-flow interpolation

$$\mathbf{x}_t = (1-t)\,\mathbf{x}_0 + t\,\mathbf{x}_1, \qquad \mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

when p_1 is isotropic Gaussian (equivalently, $\mathbf{x}_1 \equiv \boldsymbol{\epsilon}$). Flow matching Lipman et al. [2023] trains a velocity field $\mathbf{u}_t$ by regressing conditional vector fields along these paths; rectified-flow training

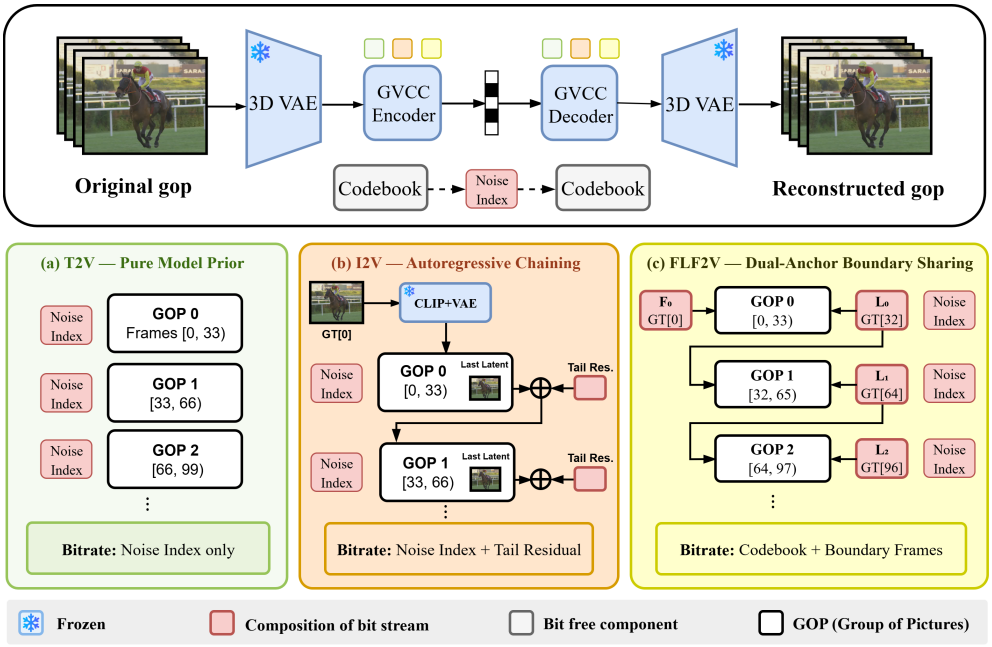

Figure 2: **Overview of the GVCC framework.** Top: shared pipeline—a frozen 3D VAE encodes the GOP into latent space, where GVCC compresses it into codebook noise indices; the decoder replays the same trajectory to reconstruct the video. Bottom: three conditioning strategies. (a) T2V: codebook only, no reference frame. (b) I2V: autoregressive GOP chaining with tail residual correction. (c) FLF2V: dual-anchor boundary sharing across consecutive GOPs.

then fits $\mathbf{u}_t$ to the instantaneous velocity of Eq. (3). Sampling is usually performed by integrating the probability-flow ODE backward from $\mathbf{x}_1$ to $\mathbf{x}_0$:

$$\mathrm{d}\mathbf{x}_t = \mathbf{u}_t(\mathbf{x}_t)\,\mathrm{d}t. \tag{4}$$

This process is purely deterministic: given $\mathbf{x}_1$, the trajectory is unique, leaving no per-step degree of freedom for embedding codebook bits.

3.2 ODE-to-SDE conversion

To introduce controllable noise injection in Eq. (4), we use the score-SDE construction Song et al. [2021b]: for any probability-flow ODE generating $\{p_t\}$, there exists a family of reverse-time SDEs sharing the same marginals for any diffusion coefficient $g_t \geq 0$:

$$\mathrm{d}\mathbf{x}_t = \underbrace{\left[\mathbf{u}_t(\mathbf{x}_t) - \frac{g_t^2}{2}\nabla \log p_t(\mathbf{x}_t)\right]}_{\mathbf{f}_t(\mathbf{x}_t)}\mathrm{d}t \;+\; g_t\,\mathrm{d}\bar{\mathbf{w}}, \tag{5}$$

Diffusion schedule as a bit-budget constraint. Unlike prior ODE-to-SDE conversions motivated by sample diversity [Kim et al., 2025, Ma et al., 2024], the schedule g_t in our setting is selected using a bit-budget consideration specific to codebook compression. At each Euler–Maruyama step, the stochastic innovation $g_t\sqrt{\Delta t}\,\mathbf{z}^*$ must be representable within the M-atom subspace spanned by the selected codebook entries. Under the near-orthogonality assumption of Vaisman et al. [2025], this subspace captures at most a fraction $c(M, d)$ of an arbitrary Gaussian innovation in latent dimension d, where $c(M, d)$ is monotonically increasing in M and saturates as M approaches the effective rank of the residual. In the video-latent setting, spatio-temporal correlation may weaken strict orthogonality; we therefore treat $c(M, d)$ as an empirical effective-rank quantity, with the saturation in Fig. 6(a) providing qualitative support rather than a direct estimate. The resulting design condition is

$$g_t^2\,\Delta t \;\leq\; \rho \cdot c(M, d) \cdot \mathrm{Var}(r_t), \tag{6}$$

for some safety factor $\rho = \mathcal{O}(1)$, while also vanishing at $t = 0$ to avoid spending bits on residuals below the codebook's resolution floor. We adopt

$$g_t = g_{\text{scale}} \cdot t^2 \tag{7}$$

as the simplest polynomial form satisfying both conditions. The scalar g_{scale} then serves as the primary rate–quality knob, subject to the upper bound in Eq. (6). Appendix E.1 verifies this interpretation empirically: the narrow stability window $g_{\text{scale}} \in [2, 3]$ observed in Fig. 6(d) is consistent with the bound being saturated at $M = 64$, and Fig. 6(a) confirms the corresponding saturation of $c(M, d)$ as M grows.

3.3 Score function from the velocity field

The SDE drift in Eq. (5) requires $\nabla \log p_t(\mathbf{x}_t)$, which the rectified-flow network does not output directly. For interpolants of the form Eq. (1) with learned velocity $\mathbf{u}_t$, the score admits a closed form in terms of $(\alpha_t, \sigma_t, \dot{\alpha}_t, \dot{\sigma}_t)$ Ma et al. [2024]:

$$\nabla \log p_t(\mathbf{x}_t) = \frac{1}{\sigma_t} \cdot \frac{\alpha_t\, \mathbf{u}_t(\mathbf{x}_t) - \dot{\alpha}_t\, \mathbf{x}_t}{\dot{\alpha}_t\, \sigma_t - \alpha_t\, \dot{\sigma}_t}. \tag{8}$$

Substituting the rectified-flow schedule [Eq. (2)] (i.e., $\alpha_t = 1 - t$, $\sigma_t = t$, $\dot{\alpha}_t = -1$, $\dot{\sigma}_t = 1$) yields the RF specialization

$$\nabla \log p_t(\mathbf{x}_t) = -\frac{(1 - t)\, \mathbf{u}_t(\mathbf{x}_t) + \mathbf{x}_t}{t}. \tag{9}$$

Under Eq. (3), $(1 - t)\, \mathbf{u}_t + \mathbf{x}_t$ tracks the noise coordinate $\mathbf{x}_1$, recovering the usual score–noise relation. Eq. (9) uses only the pretrained velocity network.

Substituting Eqs. (7) and (9) into the drift of Eq. (5) gives the explicit SDE drift

$$\mathbf{f}_t(\mathbf{x}_t) = \mathbf{u}_t(\mathbf{x}_t) + \frac{g_t^2}{2} \cdot \frac{(1 - t)\, \mathbf{u}_t(\mathbf{x}_t) + \mathbf{x}_t}{t}. \tag{10}$$

When $g_t = 0$, $\mathbf{f}_t$ reduces to $\mathbf{u}_t$, so the SDE strictly generalizes the RF ODE.

3.4 Codebook-driven SDE sampling

We adopt the codebook mechanism of DDCM Ohayon et al. [2025] and Turbo-DDCM Vaisman et al. [2025], adapting it from DDPM reverse noise to the SDE [Eq. (5)]. A single Euler–Maruyama (EM) step with step size Δt takes the form

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t - \mathbf{f}_t(\mathbf{x}_t)\, \Delta t + g_t \sqrt{\Delta t}\, \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{11}$$

where $\mathbf{z}$ is the per-step randomness to be replaced by codebook atoms. Following Turbo-DDCM's thresholding strategy, we replace it with a codebook-selected vector $\mathbf{z}^*$ as follows.

At each SDE step, the encoder first estimates the clean signal as $\hat{\mathbf{x}}_{0|t} = \mathbf{x}_t - t \cdot \mathbf{u}_t(\mathbf{x}_t)$ and computes the denoising residual $\mathbf{r} = \mathbf{x}_0 - \hat{\mathbf{x}}_{0|t}$, where $\mathbf{x}_0$ is the ground-truth latent available only at the encoder. A reproducible codebook $\mathcal{C} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(K)}\}$ is then generated from a deterministic seed shared by both sides. The M atoms with the largest absolute inner product with $\mathbf{r}$ are selected, along with their signs:

$$\{j_1, \ldots, j_M\} = \text{top-M}_i \left|\langle \mathbf{z}^{(i)}, \mathbf{r} \rangle\right|, \quad s_k = \text{sign}\big(\langle \mathbf{z}^{(j_k)}, \mathbf{r} \rangle\big). \tag{12}$$

The selected atoms are combined and normalized to unit variance to match the theoretical noise magnitude of the SDE:

$$\mathbf{z}^* = \frac{\sum_{k=1}^{M} s_k\, \mathbf{z}^{(j_k)}}{\text{std}\big(\sum_{k=1}^{M} s_k\, \mathbf{z}^{(j_k)}\big)}. \tag{13}$$

The noise in the codebook $\mathbf{z}^*$ then replaces the Gaussian innovation in the Euler update:

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t - \mathbf{f}_t(\mathbf{x}_t)\, \Delta t + g_t \sqrt{\Delta t}\, \mathbf{z}^*. \tag{14}$$

Since both encoder and decoder share the same seed, model weights, and codebook construction rule, transmitting only the M indices and signs per step suffices for the decoder to reproduce the identical trajectory. For the last N steps, we set $g_t = 0$ and revert to the deterministic ODE $\mathbf{x}_{t-\Delta t} = \mathbf{x}_t - \mathbf{u}_t(\mathbf{x}_t)\, \Delta t$, which requires zero transmitted bits since both sides produce identical outputs from the synchronized preceding state.

3.5 T2V: Pure Generative Prior Compression

T2V corresponds to the conditioning-free regime of GVCC: no reference frame is transmitted, the model is driven only by an empty text prompt, and the entire bitrate is allocated to codebook indices. This setting isolates the role of the pretrained video model as a spatio-temporal prior, with the codebook guiding an otherwise unconditioned generation trajectory toward the target video.

Because no spatial anchor is provided, the bitrate per GOP reduces to the codebook cost alone:

$$\mathrm{BPP}_{\mathrm{T2V}} = \frac{(T-N) \cdot F \cdot B_{\mathrm{step}}}{F_{\mathrm{px}} \times H_{\mathrm{px}} \times W_{\mathrm{px}}}, \tag{15}$$

which yields the lowest bitrate among the three variants. T2V therefore serves as a reference point for evaluating how much reconstruction quality can be obtained from the pretrained generative prior under codebook-only control.

A design trade-off of T2V is the lack of explicit spatial anchoring: without a reference frame, the decoded content may exhibit mild positional drift or appearance variation across GOPs.

3.6 I2V: Autoregressive Compression with Tail Correction

I2V conditions each GOP on a single reference frame, encoded through CLIP features and VAE latents, and therefore provides the strongest spatial anchoring among the three variants. To reduce side-information cost, we use an autoregressive GOP structure: the first GOP takes a compressed version of the source's first frame as its reference, while each subsequent GOP reuses the decoded last frame of the previous GOP:

$$\mathrm{GOP}_0\colon\ \mathrm{ref} = \tilde{\mathbf{I}}_0^{\mathrm{codec}}, \qquad \mathrm{GOP}_{n>0}\colon\ \mathrm{ref} = \tilde{\mathbf{I}}_{n-1}^{\mathrm{last}}. \tag{16}$$

We mitigate autoregressive error accumulation, which is concentrated on each GOP's last frame (both the hardest to reconstruct and the reference for the next GOP), with two complementary mechanisms.

First, *adaptive tail-frame atom allocation* increases the codebook atom count from M to M_{tail} for the last F_{tail} latent frames, allocating more bits to the temporally most difficult portion of the GOP.

Second, *tail latent residual correction* transmits a quantized residual for the final latent frame. After SDE encoding, the encoder computes the residual between the ground-truth latent and the decoded latent at the last temporal position, quantizes it to 8 bits per channel using min/max normalization, and compresses it losslessly with zlib. At the decoder, this residual is added back to the reconstructed latent before VAE decoding, improving the quality of the frame that will be propagated as the next GOP reference. The resulting overhead is included in the bitrate.

The total bitrate per GOP is:

$$\mathrm{BPP}_{\mathrm{I2V}} = \frac{(T-N) \cdot F \cdot B_{\mathrm{step}} + B_{\mathrm{tail_residual}} + \mathbb{1}_{n=0} B_{\mathrm{ref}}}{F_{\mathrm{px}} \times H_{\mathrm{px}} \times W_{\mathrm{px}}}, \tag{17}$$

where $B_{\mathrm{tail_residual}}$ and B_{ref} denote the compressed residual and initial-reference sizes in bits. The indicator charges the serialized initial reference once; later references are inherited from decoded outputs. All reported I2V bitrates include this initial-reference cost.

3.7 FLF2V: Dual-Anchor Compression with Boundary Sharing

FLF2V conditions each GOP on both its first and last frames, providing dual temporal anchors that constrain the generative trajectory from both ends. Compared with single-anchor I2V, this design is intended to limit temporal drift by requiring the model to satisfy boundary conditions at both the beginning and the end of the segment.

Both boundary frames are compressed with the `cheng2020-attn` image codec at quality level 4 in CompressAI Bégaint et al. [2020] and used to construct the conditioning signal. Specifically, both frames are encoded through CLIP and placed at the temporal endpoints of the VAE latent volume, together with a binary mask indicating the conditioned positions. The resulting generation process can be viewed as interpolation between two known endpoints, which is more constrained than extrapolation from a single anchor.

A key component of FLF2V is *boundary-sharing GOP chaining*, which amortizes boundary-frame cost across consecutive GOPs. The last frame of GOP n is reused as the first frame of GOP $n{+}1$:

$$\text{GOP}_n\colon\ \text{first} = \tilde{I}_n,\ \ \text{last} = \tilde{I}_{n+1}, \qquad \text{GOP}_{n+1}\colon\ \text{first} = \tilde{I}_{n+1},\ \ \text{last} = \tilde{I}_{n+2}. \tag{18}$$

Under this scheme, GOP 0 transmits two boundary frames, while each subsequent GOP transmits only one new boundary frame, reducing boundary-frame overhead by approximately 50% over long sequences. The bitrate is:

$$\text{BPP}_{\text{FLF2V}} = \frac{(T{-}N)\cdot F\cdot B_{\text{step}} + B_{\text{boundary}}}{F_{\text{px}}\times H_{\text{px}}\times W_{\text{px}}}, \tag{19}$$

where B_{boundary} equals $B_{\text{first}} + B_{\text{last}}$ (in bits) for the initial GOP and B_{last} for all subsequent GOPs. Thus, G consecutive GOPs contain $G + 1$ paid anchor streams; neither the initial anchor nor the shared anchors are treated as free. Adjacent GOPs are conditioned on the same decoded boundary frame at their junction.

4 Experiments

4.1 Experimental Setup

All three GVCC variants use the Wan 2.1 14B model family. Unless otherwise specified, we evaluate on all seven UVG sequences Mercat et al. [2020]. The main operating-point comparison uses 1080p; GVCC's native 1920×1088 output is resized, not cropped, to 1920×1080 for evaluation. The temporal stress test uses the *Jockey* sequence at 1280×720. Each GVCC GOP contains 33 frames. We additionally test a self-captured video to reduce concerns about possible overlap with the model's training distribution.

We report PSNR, MS-SSIM Wang et al. [2003], LPIPS Zhang et al. [2018], DISTS Ding et al. [2022], and BPP. LPIPS uses the official AlexNet model (version 0.1), and DISTS uses the original implementation of Ding et al. All local reconstructions are evaluated with the same RGB pipeline and BT.601 color conversion. GNVC-VD has no public implementation; its approximate curve readings are therefore marked with an asterisk.

For a bitstream of B bytes representing F frames, we compute $\text{BPP} = 8B/(FHW)$ from the serialized stream, including transmitted reference frames and anchors. Evaluations use each codec's GOP-aligned output window; using the common first 94 aligned frames changes mean LPIPS and DISTS by at most 0.08%. VMAF is evaluated on this common 94-frame window.

All 720p experiments are conducted on a single NVIDIA RTX 6000 Ada GPU with 48 GB memory. Full hyperparameter settings are provided in Appendix E, and the default parameter selection procedure is described in Appendix E.1.

4.2 Temporal Diagnostics Across GOPs

Table 1 reports a six-GOP, 720p stress test on the high-motion *Jockey* sequence. Rates are computed from the sequence-specific serialized streams, including transmitted anchors.

At the displayed low-rate point, DCVC-RT has lower E_{warp}, while T2V has lower tOF. Following Lai et al. Lai et al. [2018], E_{warp} uses a common RAFT-large Teed and Deng [2020] backward flow estimated on each ground-truth pair, a forward–backward consistency mask, and RGB L1 warping error. tOF follows the optical-flow deviation of TecoGAN Chu et al. [2020], and CLIP-F is adjacent-frame CLIP cosine similarity. Per-frame boundary LPIPS profiles are reported in Appendix B.

4.3 Trade-offs Among GVCC Variants

We first compare the three GVCC variants on all seven UVG sequences at 1280×720, using three 33-frame GOPs per sequence. For reference, we also evaluate DCVC-RT on the same resized sequences.

Table 2 shows the original three-way trade-off. I2V achieves the highest fidelity but spends substantially more bits on its tail residual; T2V uses only codebook indices and occupies the lowest-rate regime; FLF2V lies between them by sharing compressed boundary anchors.

Table 1: **Compact temporal diagnostics on UVG *Jockey* at 720p.** I2V's tail-residual stream is capped to a 0.050-bpp total budget.

Method	Setting	BPP	$E_{\text{warp}} \downarrow$	tOF↓	CLIP-F↑
GVCC-T2V	—	0.0048	0.0102	0.565	0.9823
GVCC-FLF2V	—	0.0096	0.0103	0.624	0.9838
GVCC-I2V	capped	0.050	0.0117	1.258	0.9830
GLC-Video	q0	0.0095	0.0179	2.949	0.9743
DCVC-RT	q0	0.0045	0.0060	1.353	0.9812

Table 2: **Comparison of GVCC variants on UVG 720p** (3 GOPs per sequence, 33 frames/GOP). DCVC-RT at avg. ∼0.005 bpp—matched to GVCC-T2V's bitrate—is included as a rate-matched perceptual reference. Best and second-best results among GVCC variants are highlighted.

	GVCC-T2V			**GVCC-I2V**			**GVCC-FLF2V**			**DCVC-RT**
Seq.	PSNR↑	LPIPS↓	BPP	PSNR↑	LPIPS↓	BPP	PSNR↑	LPIPS↓	BPP	LPIPS↓
Beau.	31.79	0.154	0.0048	32.90	0.109	0.0567	32.26	0.158	0.0075	0.442
Bosp.	30.32	0.098	0.0048	33.63	0.055	0.0546	32.72	0.086	0.0108	0.233
Honey.	30.78	0.052	0.0048	36.21	0.020	0.0543	34.73	0.033	0.0141	0.180
Jock.	31.28	0.090	0.0048	33.27	0.070	0.0534	33.09	0.080	0.0105	0.251
RSG	26.74	0.086	0.0048	29.48	0.053	0.0516	29.44	0.063	0.0168	0.282
SnD	25.55	0.239	0.0048	30.11	0.136	0.0549	29.52	0.166	0.0177	0.485
Yacht.	26.60	0.103	0.0048	28.21	0.075	0.0549	28.02	0.089	0.0141	0.315
Avg.	29.01	0.117	0.0048	**31.97**	**0.074**	0.0543	**31.40**	**0.096**	0.0131	0.313
Codebook	18.4 KB (100%)			18.4 KB (∼9%)			18.4 KB (∼40%)			—
Ref frames	0			Compressed first + AR			∼28 KB (∼60%)			—
Tail residual	N/A			∼189 KB (∼91%)			N/A			—
Spatial anchor	None			Single (first)			Dual (first + last)			Learned
GOP structure	Independent			AR chain			Boundary sharing			P-frame
Strength	Lowest bitrate			Highest fidelity			Best balance			Best PSNR

Why three variants? The variants test three ways of addressing cross-GOP consistency: no explicit anchor (T2V), an autoregressive first-frame anchor (I2V), and shared endpoint anchors (FLF2V). I2V is an exploratory attempt to reduce discontinuities through autoregression; its present residual/codebook allocation is not yet balanced, as reflected by its bitrate and boundary profile (Appendix B).

4.4 Rate–Distortion Curve Comparison

For T2V and FLF2V, increasing M moves each curve to the right while improving all three reported quality metrics, with diminishing changes toward the high-M end. T2V occupies the lowest-rate region. FLF2V is shifted right by its shared boundary anchors, while lying below T2V in LPIPS and DISTS and above it in PSNR. These local relationships do not establish a global ordering among the designs.

I2V lies in a separate rate region because its tail residual dominates the stream. Its raw high-rate points are reported in Table 8 (Appendix D) rather than extrapolating baselines across a non-overlapping BPP interval.

In the overlapping low-rate regions, the T2V and FLF2V curves lie below the evaluated fidelity-oriented baselines in LPIPS and DISTS, while the latter generally retain higher PSNR. FLF2V also occupies a lower perceptual-distortion region than the evaluated generative baselines at nearby rates. The three GVCC variants are reported as distinct curves, without extrapolation across non-overlapping rate ranges.

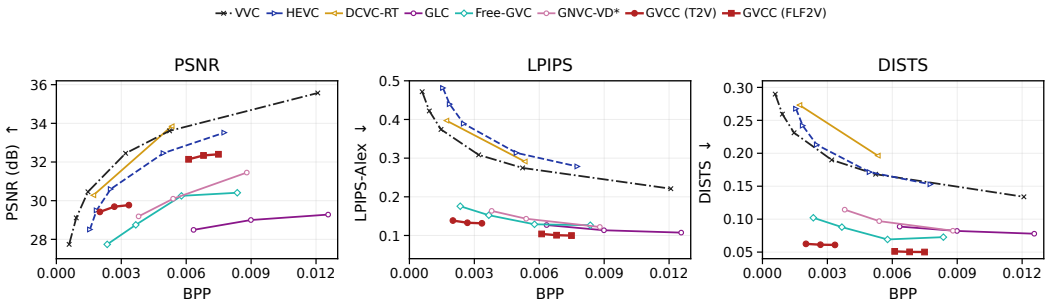

Figure 3: **Full-UVG 1080p ultra-low-rate curves.** From left to right: PSNR, LPIPS-Alex, and DISTS versus measured BPP. Only measured points below 0.013 bpp are shown, and lines connect operating points only within the same method or GVCC variant. I2V is outside this shared rate region and is reported in Table 8 (Appendix D). Free-GVC is our reproduction, and GNVC-VD* is digitized from its published curves.

HEVC and VVC use HM-16.25 and VTM-17.0, respectively; DCVC-RT and GLC-Video use released code and weights. Free-GVC is our implementation, while GNVC-VD is shown only as labeled readings from its published curves. Implementation details are listed in Appendix C.

Complementary VMAF, FVD, and spatial patch FID/KID results are provided in Appendix D.

Limitations

Computational cost. GVCC relies on a 14B-parameter video DiT Peebles and Xie [2023] with T=20 NFEs per GOP. On a single RTX 6000 Ada, encoding a 33-frame GOP takes ∼90 s and decoding ∼60 s, substantially slower than HEVC/VVC or learned predictive codecs Jia et al. [2025]. Improving runtime efficiency will likely require distillation, faster samplers, and hardware-aware optimization.

Distribution shift. Faithfulness on controlled color/text/face tests and a self-captured out-of-distribution clip is reported in Appendix F. On the OOD clip, perceptual quality (LPIPS 0.119) and color fidelity ($\Delta E = 2.77$) remain in a usable range, while PSNR is several dB lower than on in-distribution sequences—a pattern consistent with the perception-oriented optimization of generative codecs (PSNR is not the design objective). Behavior on content stylistically distant from common video distributions is therefore expected to vary, a property shared with other zero-shot generative methods rather than specific to GVCC.

Rate curves and the I2V design. The variants occupy different rate ranges, and I2V's current tail correction remains bitrate-heavy; a lighter residual design is left for future work.

Theoretical refinement under shifted-time schedules. Our diffusion coefficient $g_t = g_{\text{scale}} \cdot t^2$ is formulated in the unshifted linear-time domain, whereas Wan 2.1 uses an SD3-style Esser et al. [2024] shifted reparameterization $\tau = st/(1 + (s-1)t)$ with s=5.0. While this formulation is empirically stable across our evaluations (g_{scale}=2.0–3.0), extending the bit-budget analysis to SD3-style shifted reparameterizations is an interesting direction for tighter rate–quality control.

5 Conclusion

We presented GVCC, a zero-shot video compression framework that converts the deterministic ODE sampling of pretrained rectified-flow video models into codebook-driven stochastic decoding, with the diffusion coefficient formulated as a bit-budget-constrained design choice. T2V, I2V, and FLF2V expose different allocations of codebook, anchor, and residual bits. Their UVG rate–quality curves and temporal diagnostics demonstrate the potential of pretrained video generators as a foundation for zero-shot generative video compression.

Acknowledgments and Disclosure of Funding

This work received no dedicated third-party funding. The authors declare no competing interests.

References

Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 221–231, 2019. URL https://arxiv.org/abs/1804.02958.

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2023. URL https://arxiv.org/abs/2209.15571. ICLR 2023.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. URL https://arxiv.org/abs/2303.08797.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1802.01436.

Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. CompressAI: a PyTorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020. URL https://arxiv.org/abs/2011.03029.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. URL https://arxiv.org/abs/2311.15127.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. URL https://arxiv.org/abs/1901.07821.

Marlène Careil, Matthew J. Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.10325.

Xiangyu Chen, Jixiang Luo, Jingyu Xu, Fangqiu Yi, Chi Zhang, and Xuelong Li. Generative video compression: towards 0.01% compression rate for video transmission. *Vicinagearth*, 3(1):7, 2026.

Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics*, 2020. URL https://arxiv.org/abs/1811.09393.

Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, 2022. URL https://arxiv.org/abs/1801.07698.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. URL https://arxiv.org/abs/2004.07728.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144, 2020.

Zongyu Guo, Zhaoyang Jia, Jiahao Li, Xiaoyi Zhang, Bin Li, and Yan Lu. Generative latent video compression. *arXiv preprint arXiv:2510.09987*, 2025. URL https://arxiv.org/abs/2510.09987.

Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7033–7042, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.11239.

Emiel Hoogeboom, Eirikur Agustsson, Fabian Mentzer, Luca Versari, George Toderici, and Lucas Theis. High-fidelity image compression with score-based generative models. *arXiv preprint arXiv:2305.18231*, 2023. URL https://arxiv.org/abs/2305.18231.

Zhihao Hu, Guo Lu, and Dong Xu. FVC: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1502–1511, 2021.

Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu. Towards practical real-time neural video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12543–12552, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Jia_Towards_Practical_Real-Time_Neural_Video_Compression_CVPR_2025_paper.html.

Zhaoyang Jia, Zihan Zheng, Naifu Xue, Jiahao Li, Bin Li, Zongyu Guo, Xiaoyi Zhang, Houqiang Li, and Yan Lu. Cod: A diffusion foundation model for image compression, 2026. URL https://arxiv.org/abs/2511.18706.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2206.00364.

Muhammad Umar Karim Khan, Aaron Chadha, Mohammad Ashraful Anam, and Yiannis Andreopoulos. Perceptual video compression with neural wrapping. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17743–17754, 2025.

Jaihoon Kim, Taehoon Yoon, Jisung Hwang, and Minhyuk Sung. Inference-time scaling for flow models via stochastic generation and rollover budget forcing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European Conference on Computer Vision*, pages 170–185, 2018.

Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26099–26108, 2024.

Xiaoyue Ling, Chuqin Zhou, Chunyi Li, Yunuo Chen, Yuan Tian, Guo Lu, and Wenjun Zhang. Free-GVC: Towards training-free extreme generative video compression with temporal coherence. *arXiv preprint arXiv:2602.09868*, 2026. URL https://arxiv.org/abs/2602.09868.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. URL https://arxiv.org/abs/2209.03003.

Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11006–11015, 2019.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024. URL https://arxiv.org/abs/2401.08740.

Qi Mao, Hao Cheng, Tinghan Yang, Libiao Jin, and Siwei Ma. Generative neural video compression via video diffusion prior. *arXiv preprint arXiv:2512.05016*, 2025. URL https://arxiv.org/abs/2512.05016.

Fabian Mentzer, George Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2006.09965.

Fabian Mentzer, Eirikur Agustsson, Johannes Ballé, David Minnen, Nick Johnston, and George Toderici. Neural video compression using gans for detail synthesis and propagation. In *European Conference on Computer Vision*, pages 562–578. Springer, 2022.

Alexandre Mercat, Marko Viitanen, and Jarno Vanne. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 297–302, 2020.

David Minnen, Johannes Ballé, and George D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1809.02736.

Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 47044–47089. PMLR, 2025. URL https://proceedings.mlr.press/v267/ohayon25a.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

Linfeng Qi, Zhaoyang Jia, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Generative latent coding for ultra-low bitrate image and video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. Lossy image compression with foundation diffusion models. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2404.08580.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*, 25:7311–7322, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2010.02502.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2021b. URL https://arxiv.org/abs/2011.13456.

Xun Su and Hiroyuki Kasai. Noise is all you need: Solving linear inverse problems by noise combination sampling with diffusion models, 2025. URL https://arxiv.org/abs/2510.23633.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, 2020.

Lucas Theis, Tim Salimans, Matthew D. Hoffman, and Fabian Mentzer. Lossy compression with gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022. URL https://arxiv.org/abs/2206.08889.

Amit Vaisman, Guy Ohayon, Hila Manor, Michael Elad, and Tomer Michaeli. Turbo-DDCM: Fast and flexible zero-shot diffusion-based image compression. *arXiv preprint arXiv:2511.06424*, 2025. URL https://arxiv.org/abs/2511.06424.

Jeremy Vonderfecht and Feng Liu. Lossy compression with pretrained diffusion models. *arXiv preprint arXiv:2501.09815*, 2025. URL https://arxiv.org/abs/2501.09815.

Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. IEEE, 2003.

WanTeam, Ang Wang, Baole Ai, Bin Wen, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2209.06950.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.06072.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. URL https://arxiv.org/abs/1801.03924.

A Conceptual Foundations of GVCC

Section 3 derives GVCC formally; this appendix offers a complementary conceptual reading of how the SDE+codebook construction operates at bitrates down to approximately 0.003 bpp in our measured configuration.

A.1 A coding channel via marginal-preserving SDE

The rectified-flow ODE in Eq. (4) is fully deterministic, so the only quantity through which a bitstream could influence reconstruction is $\mathbf{x}_T$—prohibitively expensive to transmit at any nontrivial precision. SDE conversion (Eq. (5)) introduces a per-step diffusion term whose marginals coincide with the ODE's by construction [Song et al., 2021b]: it does not change *what* the model generates in expectation, only *how* the trajectory is realized. Each step now exposes a controllable perturbation through which the bitstream can specify the reconstruction. Prior ODE-to-SDE conversions [Kim et al., 2025, Ma et al., 2024] use this construction to widen sample diversity; GVCC repurposes the identical machinery to open a coding channel, with g_t acting as a bit-budget knob (Eq. (6)). Although the SDE is mathematically stochastic, in GVCC the executed trajectory is deterministic for both encoder and decoder: the randomness in $g_t \, \mathrm{d}\bar{\mathbf{w}}$ is replaced by codebook-selected innovations $\mathbf{z}^*$ uniquely determined by the transmitted indices and the shared seed.

A.2 Bits provide fidelity, prior provides plausibility

The Euler–Maruyama update of Eq. (14) decomposes each step into two terms with complementary roles:

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t \underbrace{-\mathbf{f}_t(\mathbf{x}_t)\,\Delta t}_{\text{drift (plausibility)}} + \underbrace{g_t\sqrt{\Delta t}\,\mathbf{z}^*}_{\text{noise (fidelity)}}.$$

The drift is *source-agnostic*: computed from $\mathbf{u}_t$ and $\mathbf{x}_t$ alone, with no access to $\mathbf{x}_0$, it keeps the trajectory on the manifold of natural videos learned by the model. The noise is *source-driven*: the codebook-selected $\mathbf{z}^*$ is constructed from the residual $\mathbf{r} = \mathbf{x}_0 - \hat{\mathbf{x}}_{0|t}$ and accumulates over the $T - N$ noise-injection steps (cf. Sec. 3.4) to steer the trajectory toward $\mathbf{x}_0$. Neither term suffices alone: pure drift yields a plausible video unrelated to $\mathbf{x}_0$; pure noise escapes the manifold.

This division motivates the intended rate–perception behavior. Conventional codecs must represent both visual structure and source-specific content, whereas GVCC delegates part of the plausibility prior to the frozen generator and uses transmitted indices to localize the reconstruction. Whether this mechanism improves a rate–distortion or rate–perception curve must be judged from matched multi-rate experiments; the present full-UVG results are reported as measured BPP curves without a global BD claim.

B GOP-Boundary Profiles

Consecutive-frame profiles. Figure 4 visualizes consecutive-frame MAE across GOPs. T2V shows periodic boundary spikes on *HoneyBee* and *Beauty*, while I2V shows sharp boundary changes on *Jockey* associated with its tail correction.

Boundary LPIPS. We report per-frame LPIPS statistics around GOP boundaries. For each seam at output index s, the boundary set contains frames $\{s-1, s\}$. "Intra" contains every other frame, Δ_{boundary} is boundary minus intra LPIPS, and the standard deviation is computed over the complete per-frame LPIPS profile. The same LPIPS-AlexNet protocol from Sec. 4.1 is used throughout. The seam indices are $\{33, 66, 99, 132, 165\}$ for T2V/I2V and $\{33, 65, 97, 129, 161\}$ for FLF2V.

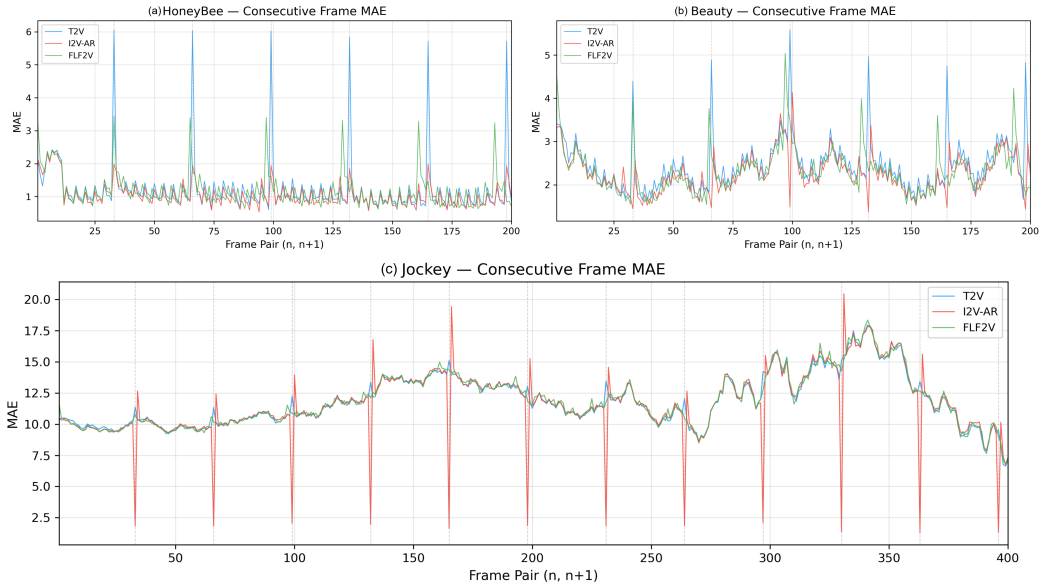

Figure 4: Consecutive-frame MAE on *HoneyBee*, *Beauty*, and *Jockey*. Dashed vertical lines indicate GOP boundaries.

Table 3: Boundary LPIPS statistics on *Jockey* at 720p. I2V uses the same capped stream as Table 1.

Variant	BPP	Frames / seams	Intra	Boundary	Δ / std.
T2V	0.00483	198 / 5	0.0880	0.0898	+0.0018 / 0.0048
FLF2V	0.009591	193 / 5	0.0780	0.0856	+0.0076 / 0.0056
I2V	0.050 (capped)	198 / 5	0.0712	0.1147	+0.0435 / 0.0217

C Codec and Metric Implementations

Table 4 summarizes codec and metric implementations. GNVC-VD is the only curve-read result; the remaining plotted methods use local outputs and a unified metric pipeline.

Table 4: Codec and metric implementations used in evaluation.

Component	Version and configuration
HEVC / VVC	HM-16.25 / VTM-17.0, low-delay configuration with intra period 32.
DCVC-RT / DCVC-FM	Official CVPR 2025 / 2024 weights; RT uses serialized streams and FM uses entropy-model rate estimates.
GLC-Video	Official released code and weights.
Free-GVC	Our implementation of the published method using CogVideoX-2B.
GNVC-VD	Approximate readings from the published RD curves; not locally recomputed.
LPIPS / DISTS	Official LPIPS-Alex v0.1, and the DISTS metric of Ding et al. as implemented in PIQ 0.8.
Additional metrics	VMAF v0.6.1, FVD with Kinetics-400 I3D, patch FID/KID with Inception-v3, RAFT-large optical flow, and CLIP ViT-B/32.

Licenses. The pretrained Wan2.1 models and CogVideoX-2B, used for our Free-GVC implementation, are released under the Apache-2.0 license. UVG is distributed under CC BY-NC 3.0 (copyright Digiturk). HM and VTM are distributed under the BSD 3-Clause license, DCVC-RT and DCVC-FM under the MIT license, GLC-Video under Apache-2.0, and CompressAI under the BSD 3-Clause Clear license. For the metrics, the `lpips` package is BSD-2-Clause, PIQ (DISTS) and pytorch-fid (Inception-v3) are Apache-2.0, VMAF is BSD-2-Clause-Patent, the Kinetics-400 I3D model is Apache-2.0, RAFT is BSD-3-Clause, and CLIP is MIT. Face identity uses the ArcFace model of In-

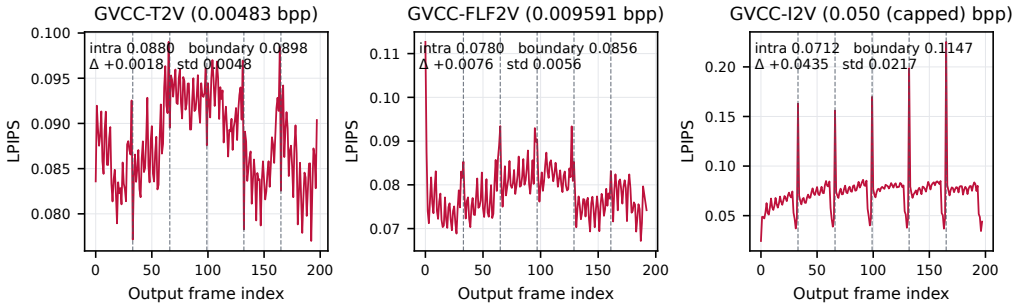

Figure 5: Per-frame LPIPS profiles used for Table 3. Dashed lines mark GOP seam indices. The complete arrays, exact boundary indices, and evaluation code are included in the supplementary material.

sightFace, whose code is MIT-licensed and whose pretrained models are provided for non-commercial research. The color, text, and face tests in Appendix F use clips from public benchmarks, and the OOD clip was captured by the authors. All assets were used for non-commercial research under these terms.

D Additional Perceptual Metrics and I2V Results

VMAF. We evaluate the first 94 aligned frames of every UVG sequence at 1080p using `vmaf_v0.6.1` and a common BT.601 RGB-to-YUV420 conversion. Scores are averaged per sequence and then equally across the seven sequences.

Table 5: VMAF on UVG 1080p. BPP denotes the measured operating point; no rate interpolation is applied.

Method	Setting	BPP	VMAF↑	Method	Setting	BPP	VMAF↑
DCVC-RT	q42	0.01835	84.59	GLC-Video	q42	0.01256	44.90
VVC	QP32	0.01208	80.67	GVCC-T2V	—	0.00268	44.25
GVCC-FLF2V	—	0.00680	67.80	GLC-Video	q21	0.00899	42.87
GVCC-I2V	—	0.05230	67.59	VVC	QP45	0.00146	42.02
HEVC	QP37	0.00776	66.68	Free-GVC	$t^* = 250$	0.00578	41.83
DCVC-RT	q21	0.00533	65.88	HEVC	QP45	0.00251	40.51
GLC-Video	q63	0.01644	46.10	GLC-Video	q0	0.00634	39.79

Despite its higher bitrate and lower LPIPS, I2V obtains nearly the same VMAF as FLF2V (67.59 vs. 67.80). This suggests that its tail residual improves perceptual detail without uniformly improving reference-aligned fidelity, highlighting the current imbalance between residual and codebook bits.

FVD. We additionally report a descriptive FVD comparison on 233 identical, non-overlapping 16-frame clips from all seven UVG sequences at 1080p. We use the first 576 frames of each long sequence and the first 272 frames of *ShakeNDry*, a fixed 400-dimensional Kinetics-400 I3D representation, and one Gaussian pooled over all clips. Every method is evaluated from lossless reconstructions using the same clip indices. The operating points are native and not rate matched; DCVC-RT uses serialized-stream BPP, GLC-Video uses its released entropy-model estimate, and GVCC includes all transmitted boundary anchors.

Patch FID and KID. We evaluate 18,816 non-overlapping 256×256 patches from the first 96 frames of all seven UVG sequences using the same TorchMetrics 1.9.0 Inception-v3 pipeline. These metrics characterize spatial patch statistics and do not measure temporal coherence.

I2V high-rate operating points.

Table 6: Descriptive FVD results on the fixed 233-clip UVG support. The leave-one-sequence-out (LOO) range is reported as a sensitivity check. Since the rates differ and the 400-dimensional covariance is rank deficient at this sample count, no cross-method efficiency or distribution-level superiority is inferred.

Method	Setting	BPP	FVD↓	LOO range
GVCC-FLF2V	$M = 80$	0.006895	40.41	40.26–49.63
GLC-Video	$q_I = 3, q_P = 63$	0.016982	41.64	41.25–51.43
DCVC-RT	$q_I = q_P = 63$	0.061345	9.69	4.87–11.80

Table 7: Patch-level FID/KID on UVG 1080p under a unified local evaluation protocol.

Method	Setting	BPP	FID↓	KID↓
GVCC-FLF2V-14B	Local	0.0070	**14.92**	**0.00216**
Free-GVC	$t^* = 215$	0.0072	18.36	0.00304 ± 0.00031

I2V residual budget. We vary the tail-residual budget on UVG *Jockey* at 1080p while keeping the remaining I2V configuration fixed.

The residual contribution is therefore not a smooth rate-control knob in the current I2V design; I2V is treated as an exploratory variant.

E Hyperparameter Configuration

Table 10 lists the full set of hyperparameters used across all three GVCC variants at 720p and 1080p resolutions.

E.1 Default Parameter Selection

We sweep each hyperparameter individually on the UVG *Beauty* sequence using T2V-1.3B at 720p, varying one parameter at a time while holding the others at default values. Results are shown in Fig. 6.

Atom count M (Fig. 6a). M is the primary bitrate control variable, with BPP scaling nearly linearly from 0.0012 (M=16) to 0.0192 (M=256). Increasing M from 16 to 64 raises PSNR by 1.2 dB at 0.0048 BPP. Beyond M=128, the measured change is smaller—M=256 adds 0.3 dB while LPIPS changes from 0.117 to 0.121. Encoding time is approximately 91 s across these settings. We use M=64 as the default in subsequent experiments.

Codebook size K (Fig. 6b). K determines the size of the per-step search space. Increasing K from 1024 to 16384 changes PSNR from 31.04 to 31.68 dB and LPIPS from 0.141 to 0.122, while encoding time changes from 46 to 90 s. Increasing to K=65536 gives 0.23 dB additional PSNR at 237 s. We therefore use K=16384 for the reported experiments.

Sampling steps T (Fig. 6c). The number of SDE steps has a large measured effect. At T=5, PSNR is 19.3 dB and LPIPS is 0.576; LPIPS changes from 0.514 to 0.211 between T=10 and T=15. From T=20 to T=30, PSNR increases by 0.5 dB while encoding time changes from 90 to 136 s. We use T=20 as the reported default.

Diffusion scale g_{scale} (Fig. 6d). This parameter exhibits a notably narrow effective range. g_{scale}=2.0 and 3.0 perform nearly identically (PSNR 31.64 vs. 31.68, LPIPS 0.122 vs. 0.122). However, quality collapses rapidly outside this window: g_{scale}=5.0 drops PSNR to 29.3 dB with LPIPS 0.597, and g_{scale}=8.0 gives 22.9 dB / 0.797. This behavior is consistent with the bit-budget interpretation of Eq. (6): increasing g_{scale} scales the innovation variance $g_t^2\,\Delta t$ quadratically, and the fixed M=64 atom set eventually becomes insufficient for residual matching. The atom-count sweep in Fig. 6(a) likewise shows diminishing changes beyond M=128. We use g_{scale}=3.0 in the reported configuration.

Table 8: Raw full-UVG 1080p I2V and DCVC-RT operating points. I2V BPP includes the initial reference frame.

Method	Setting	BPP	PSNR ↑	LPIPS ↓	DISTS ↓
I2V	$M = 60$	0.05163	32.72	0.0859	0.0501
I2V	$M = 80$	0.05230	32.83	0.0840	0.0497
I2V	$M = 100$	0.05297	32.86	0.0835	0.0496
DCVC-RT	q42	0.01835	36.94	0.2149	0.1420
DCVC-RT	q63	0.06289	39.64	0.1661	0.1071

Table 9: I2V tail-residual budget sweep on *Jockey* at 1080p.

Residual budget	BPP	PSNR↑	LPIPS↓	Status
Full	0.050	32.83	0.084	Stable
80%	0.041	30.12	0.108	Stable
50%	0.027	27.53	0.172	Stable
0%	∼0.005	—	—	Unstable from GOP 3

GOP length (Fig. 6e). A GOP of 17 frames is too short for the video model to fully exploit its temporal modeling capacity, resulting in substantially worse perceptual quality (LPIPS 0.224 vs. 0.122 at 33 frames). Extending to 49 frames yields marginal improvement (LPIPS 0.118, PSNR +0.27 dB) at slightly lower BPP (0.00471 vs. 0.00483), since the VAE's $4\times$ temporal stride compresses longer sequences more efficiently. We adopt 33 frames as the default, noting that 49 frames is a viable option for 1080p where the additional compute is justified.

Summary. The validated default configuration is M=64, K=16384, T=20, g_{scale}=3.0, GOP=33. The complete parameter table with 720p and 1080p settings is provided in Table 10. These sweeps are mainly used to determine stable and interpretable default trends, and a sanity check has also been done on the main model.

F Faithfulness and Robustness

A central concern for generative codecs is whether they reconstruct the source faithfully or merely produce plausible alternatives. To evaluate this, we conduct four targeted tests using GVCC-FLF2V at 720p (3 GOPs, 97 frames per video), focusing on color fidelity, text fidelity, face identity, and out-of-distribution (OOD) behavior. Results are summarized in Table 11.

Controlled fidelity tests. The color-chart and text-overlay tests yield $\Delta E = 0.26$ in both cases, indicating imperceptible color deviation. For text, GVCC additionally achieves PSNR 38.87 dB and LPIPS 0.002, indicating faithful reconstruction of high-contrast symbolic content under controlled in-distribution conditions.

Face identity. We measure ArcFace cosine similarity on 8 sampled frames and obtain a mean of 0.857 ($\sigma = 0.054$, range 0.765–0.937), with all sampled frames preserving the correct number of detected faces. This indicates moderate but not severe identity degradation. GOP-level PSNR remains stable across the chain (31.1→32.0→31.7 dB) and ΔE stays at 2.34, suggesting no clear evidence of cumulative identity drift across the three chained GOPs.

Out-of-distribution behavior. We additionally evaluate a self-captured clip center-cropped and resized to 720p. Perceptual quality and color fidelity remain in a usable range (LPIPS 0.119, MS-SSIM 0.910, $\Delta E = 2.77$), while PSNR is several dB lower than on the controlled in-distribution tests; this gap is consistent with the perception-oriented optimization of generative codecs and is more pronounced in PSNR than in perceptual or color metrics. Boundary-frame overhead rises modestly to about 66% of the per-GOP bitstream (108/163 KB), versus roughly 60% on the in-distribution setting, reflecting content-driven entropy variation in the learned image codec.

Summary. Across all four tests, per-GOP PSNR remains within roughly ± 1.5 dB across the chain, consistent with the boundary-sharing design of FLF2V (Sec. 3.7). The OOD clip declines

Table 10: Default hyperparameters for GVCC (720p / 1080p).

Parameter	720p	1080p	Role
M	64	80	Atoms per step (bitrate knob)
M_{tail} (I2V)	128	128	Atoms for AR tail frames
K	16384	16384	Codebook size
T	20	20	Total sampling steps
N	3	3	Bit-free ODE tail steps
g_{scale}	3.0	3.0	SDE diffusion coefficient
GOP length	33	33	Frames per GOP ($4k+1$)
Ref quality (FLF2V)	4	4	CompressAI quality level
Tail residual (I2V)	8-bit	8-bit	Quantization for AR correction
Seed	42	42	Shared encoder/decoder seed

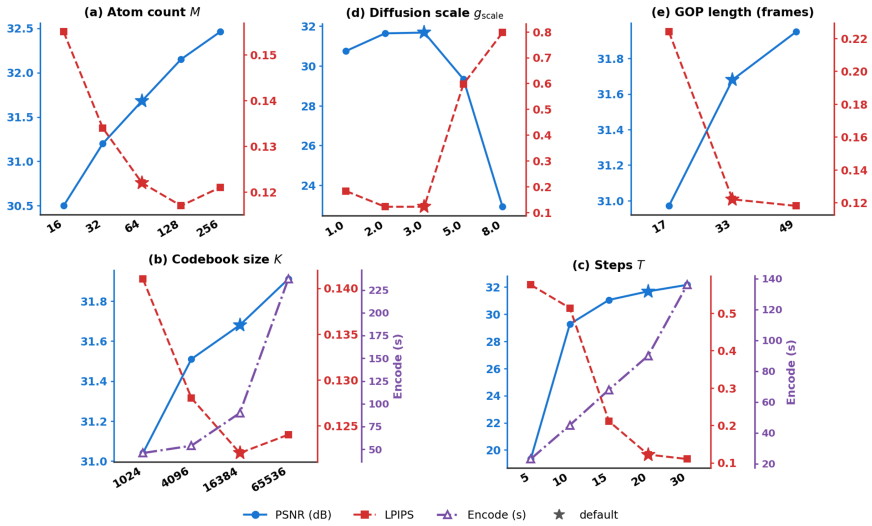

Figure 6: **Hyperparameter sweeps on UVG *Beauty* (T2V-1.3B, 720p).** Blue: PSNR (↑). Red: LPIPS (↓). Purple: encoding time. Stars: selected defaults. (a) Atom count M: quality saturates around M=64 while BPP grows linearly. (b) Codebook size K: diminishing returns beyond 16384 at rapidly increasing cost. (c) Steps T: marked degradation at T=5 and smaller changes beyond T=20. (d) Diffusion scale g_{scale}: narrow sweet spot at 2.0–3.0; collapse at higher values. (e) GOP length: 17 frames insufficient; 33 and 49 comparable.

by 1.3 dB over three GOPs (25.9→24.9→24.6 dB), while the face test varies non-monotonically (31.1→32.0→31.7 dB).

F.1 VAE Upper Bound Analysis

To disentangle the quality loss introduced by the 3D VAE from that of the codebook compression, we compare the VAE encode–decode upper bound (no GVCC, just VAE round-trip) against the full FLF2V pipeline on the four faithfulness test videos. Results are shown in Table 12.

The additional loss from GVCC beyond the VAE upper bound is small on the controlled in-distribution tests (text: −1.59 dB; color: −4.03 dB)—the VAE itself retains 40–52 dB PSNR—suggesting that the dominant degradation comes from the codebook-driven compression stage rather than from the VAE alone. For the face test, the VAE retains very high identity similarity (cosine similarity 0.987), while the full pipeline drops to 0.857, indicating that the remaining identity shift is primarily associated with the codebook-driven generative reconstruction process. The largest gap appears on the out-of-distribution clip (−5.71 dB). Notably, the VAE round-trip on this clip (30.86 dB) is markedly lower than on the controlled in-distribution tests (40+ dB), indicating that a substantial portion of the OOD degradation originates from latent reconstruction rather than from the codebook

Table 11: **Faithfulness and robustness evaluation** (GVCC-FLF2V, 720p, 3 GOPs). ΔE: CIEDE2000 color difference ($<$1: imperceptible, 1–3: barely noticeable, $>$3: clearly visible). Face ID: ArcFace Deng et al. [2022] cosine similarity between original and reconstructed faces.

Test	PSNR↑	LPIPS↓	MS-SSIM↑	ΔE↓	Face ID↑
Color	48.60	0.004	0.998	0.26	—
Text	38.87	0.002	0.998	0.26	—
Face	31.58	0.022	0.986	2.34	0.857
OOD	25.15	0.119	0.910	2.77	—

Table 12: VAE upper bound vs. FLF2V compression (720p).

Test	VAE PSNR	FLF2V PSNR	Δ (GVCC loss)
Color	52.63	48.60	-4.03
Text	40.46	38.87	-1.59
Face	35.01	31.58	-3.43
OOD	30.86	25.15	-5.71

stage alone. Increasing the atom count M may further reduce the codebook approximation error and narrow this remaining gap, subject to the saturation of $c(M, d)$ observed in Fig. 6(a).

G Rate–Distortion Sweep

To characterize GVCC's rate–distortion behavior over a wide operating range, we sweep 11 (M, K) configurations using T2V-1.3B at 480p (7 UVG sequences, 3 GOPs each). Results are shown in Table 13 and Fig. 7. BPP and PSNR increase monotonically from 0.0008 bpp / 22.5 dB to 0.0496 bpp / 30.0 dB, confirming smooth and predictable bitrate control across nearly two orders of magnitude.

Table 13: T2V-1.3B R-D sweep on UVG 480p (7 seq. $\times$ 3 GOPs).

M	K	PSNR (dB)	BPP	kbps
8	256	22.50	0.00081	5.3
16	512	24.32	0.00183	11.9
16	1024	24.76	0.00201	13.1
32	2048	26.32	0.00438	28.5
32	4096	26.62	0.00474	30.9
64	4096	27.61	0.00948	61.7
64	16384	28.06	0.01095	71.2
128	16384	28.94	0.02187	142.4
128	65536	29.28	0.02478	161.4
256	16384	29.64	0.04374	284.9
256	65536	30.00	0.04956	322.8

H Supplementary Perceptual Preference Study

To complement the distortion and perceptual metrics reported in the main text, we conducted a small-scale internal preference study. Because the study used a convenience sample from our laboratory, we report it as supplementary qualitative evidence rather than a primary claim.

Setup. Ten volunteer participants familiar with video processing research evaluated 7 clips from the UVG-1080p evaluation set. For each trial, participants viewed reconstructions from two methods side by side at approximately matched or nearby bitrate, with left/right ordering randomized. They were asked to select the version they preferred in terms of overall visual quality—including sharpness, temporal consistency, and naturalness—or to indicate no preference. Participants could replay each pair freely. Method identities were not revealed during evaluation. Because no implementation of GNVC-VD is publicly available, its stimuli were taken from the figures of the GNVC-VD paper rather than decoded locally, so this comparison is less controlled than the one against DCVC-RT.

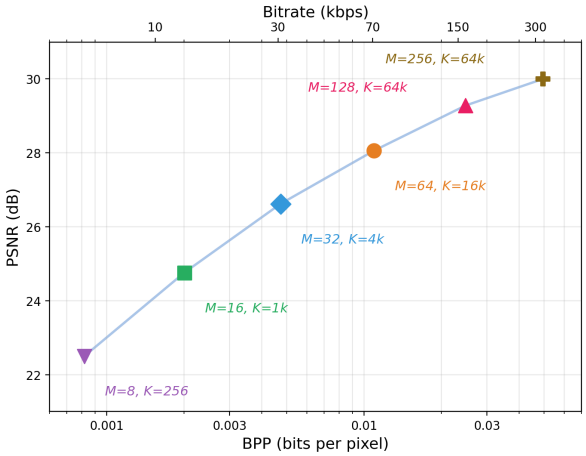

Figure 7: **Rate–distortion curve of GVCC-T2V** (1.3B, 480p, UVG average). Each point corresponds to a (M, K) configuration from Table 13. The curve spans from 5.3 kbps to 322.8 kbps with monotonically increasing quality.

Instruction shown to participants. The following instruction was shown before the study:

> You will see two reconstructed videos of the same source content shown side by side. Please select the version you prefer in terms of overall visual quality. You may consider factors such as sharpness, temporal smoothness, and visual naturalness. If needed, you may replay the pair before making your decision. If you do not observe a meaningful difference, you may choose "No preference."

Results. Table 14 reports the fraction of pairwise comparisons in which GVCC (T2V) was preferred, excluding ties. GVCC was preferred in 97% of comparisons against DCVC-RT and 88% against GNVC-VD. These preferences are consistent with the qualitative and perceptual trends reported in the main text.

Table 14: Pairwise perceptual preference (GVCC-T2V win rate, ties excluded). *GNVC-VD stimuli and rate are taken from its published figures.

Comparison	BPP (ours / baseline)	GVCC preferred
GVCC vs. DCVC-RT	∼0.005 / ∼0.005	97%
GVCC vs. GNVC-VD	∼0.005 / nearby*	88%

Ethics note. Participants were unpaid volunteers, and no identifying information was collected. This study was a minimal-risk internal perceptual evaluation intended only as supplementary evidence.

I Broader Impacts

By reducing bandwidth and storage costs, GVCC may benefit low-bandwidth deployments and energy-efficient video infrastructure. As a generative codec, it can hallucinate plausible but inaccurate content at extreme bitrates (Appendix F), making it unsuitable for forensic or evidentiary use; outputs should be flagged as generative reconstructions in such contexts.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the scope of the work: a zero-shot codec that turns a pretrained rectified-flow video model into a codebook-driven stochastic decoder, three conditioning variants (T2V, I2V, FLF2V), and evaluation on the seven UVG sequences. Section 3 describes the method. Section 4 and Appendices B–G provide the evidence, and the Limitations section states the computational cost, the behavior under distribution shift, and the rate range of each variant.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated Limitations section, placed before the Conclusion, discusses computational cost, behavior under distribution shift (with evidence in Appendix F), the bitrate-heavy tail correction of I2V, and the formulation of the diffusion coefficient under shifted time schedules. Appendix B reports behavior at GOP boundaries.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [No]

   Justification: The paper contains no formal theorems. The ODE-to-SDE conversion and the score expression (Sections 3.2 and 3.3) specialize results of Song et al. [2021b] and Ma et al. [2024], which are cited where they are used. The bit-budget condition in Eq. (6) is a design rationale with an $\mathcal{O}(1)$ safety factor rather than a proved bound, and it is checked empirically in Appendix E.1. A rigorous treatment, including shifted time schedules, is left to future work, as noted in the Limitations section.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 4.1 specifies the backbone, datasets, resolutions, GOP settings, and metrics. Appendix C describes the codec and metric implementations, and Appendix E lists all hyperparameters and how the defaults were chosen. Encoding and decoding share the seed and model weights, and the released code provides run commands for all three variants.

   Guidelines:

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

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code is publicly available at `https://github.com/CCdydy/GVCC` under the Apache-2.0 license, with installation instructions, model-weight and dataset setup, and run commands for T2V, I2V, and FLF2V. The main experiments use the public UVG dataset. Baseline codecs are run with their official releases, as listed in Appendix C.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: GVCC is zero-shot and involves no training, so there are no data splits or optimizer settings to report. Section 4.1, Appendix C, and Appendix E give all inference-time settings and explain how the default parameters were selected.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We do not report confidence intervals or significance tests. Results are computed with a fixed seed shared by the encoder and decoder, and variability across seeds is not measured. Per-sequence results are reported in Table 2, and Appendices B and D report standard deviations for the boundary profiles and for KID.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 4.1 states that the 720p experiments run on a single NVIDIA RTX 6000 Ada GPU with 48 GB of memory. The Limitations section reports about 90 s of encoding and 60 s of decoding per 33-frame GOP. The full project, including preliminary experiments, used several hundred GPU-hours.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics, and the research conforms to it. The main experiments use public benchmark videos. The small perceptual preference study involved unpaid volunteers and collected no identifying information. The Broader Impacts appendix discusses the risk of plausible but inaccurate reconstructions.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: The Broader Impacts appendix discusses lower bandwidth and storage costs as potential benefits. It also discusses the risk that generative reconstructions contain plausible but inaccurate content at extreme bitrates, which makes them unsuitable for forensic or evidentiary use, and recommends flagging outputs as generative reconstructions.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: We release no new generative model weights and no scraped dataset. The released code uses existing pretrained Wan2.1 models, which remain subject to their original licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All external assets are cited where they are used. Table 4 in Appendix C lists the codec and metric implementations with their versions, and the Licenses paragraph that follows names the license of each asset, including the pretrained models, the UVG dataset, the baseline codecs, and the metric implementations. The faithfulness tests use public benchmark clips and one clip captured by the authors.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The released code is documented in its README, which covers installation, data preparation, and run commands, and it is distributed under the Apache-2.0 license. No new dataset or model weights are released.

    Guidelines:

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

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

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

Answer: [Yes]

Justification: Appendix H gives the full instruction text shown to participants, the evaluation procedure, and the compensation. The participants were ten unpaid volunteers from our laboratory, and no identifying information was collected.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

Answer: [No]

Justification: The study in Appendix H was a minimal-risk perceptual preference evaluation in which ten unpaid volunteers from our laboratory compared reconstructions of public UVG videos. No identifying information was collected, and no invasive or high-risk procedure was involved. Formal IRB review was not sought given the minimal risk and the supplementary role of the study.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs were used only for language editing and formatting. They were not part of the method, the experiments, or the analysis.

Guidelines:

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