arXiv:2609.02291v1 [cs.CV] 2 Sep 2026

VoRTeC: Taming Foundation Flow for One-step Real time Video Compression

A Preprint

Yichong Xia
Shenzhen International Graduate School
Tsinghua University
xiayc23@tsinghua.edu.cn

Qinhong Wu
Harbin Institute of Technology, Shenzhen
2024111108@stu.hit.edu.cn

Bin Chen*
Department of Computer Science
Harbin Institute of Technology, Shenzhen
chenbin2021@hit.edu.cn

Jinpeng Wang
Department of Computer Science
Harbin Institute of Technology, Shenzhen
wjp20@tsinghua.edu.cn

Zeyuan Chen
School Of Computer Science
Peking University
chenzy01@stu.pku.edu.cn

Haoqian Wang
Shenzhen International Graduate School
Tsinghua University
wanghaoqian@tsinghua.edu.cn

September 3, 2026

Abstract

Ultra-low bitrate video compression still faces critical challenges: traditional neural video compression inevitably introduces blurring artifacts, while diffusion-based generative video compression suffers from excessive decoding latency and poor temporal consistency. To address these issues, we propose VoRTeC, a Video Compression framework built upon a foundational flow model (Wan2.1). By compactly encoding latent video representations, predicting the positions of compressed representations along flow trajectories, and integrating multi-scale priors, VoRTeC enables the compressor to harness generative video flow priors effectively. Without accessing the parameters or gradients of flow matching networks, our framework achieves one-step decoding and reconstructions with high perceptual fidelity. Meanwhile, we maintain consistency across frame groups via tail-frame reuse and prior caching. Extensive experiments demonstrate that our method reduces bit consumption by 58% compared to prior diffusion-based approaches, with decoding speed boosted by 3 to 197 times: VoRTeC achieves a decoding speed of 13 FPS at 720p and 32 FPS at 480p. Project page: https://darc8-sun.github.io/VoRTec_compress/.

1 Introduction

Video compression underpins virtually every modern visual communication pipeline, from real-time streaming and video conferencing to cloud storage and autonomous driving [Sullivan et al., 2012, Bross et al., 2021]. As bandwidth demands continue to escalate, the ability to reconstruct perceptually faithful video at extremely low bitrates has become a critical yet unresolved challenge.

Traditional hybrid codecs such as HEVC [Sullivan et al., 2012] and VVC [Bross et al., 2021] rely on block-wise prediction and transform coding, which produce visible blocking and ringing artifacts under aggressive quantization. Recent neural video codecs (NVCs) [Lu et al., 2019, Li et al., 2021, 2023, 2024, Jia et al., 2025] have surpassed these

*Correspoding Author

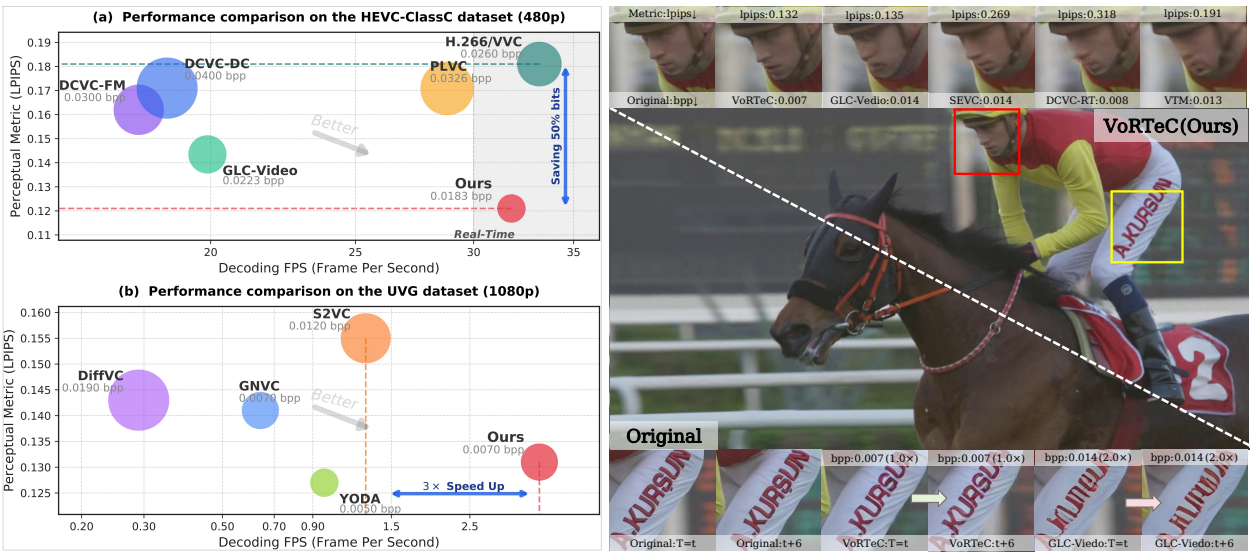


Figure 1: Left: Comparison of rate-perception performance and decoding efficiency. `VoRTeC` (Ours) delivers a better rate–perception trade-off than prior diffusion-based video codecs and enables real-time decoding, running up to 3× faster than existing diffusion-based methods. Right: Qualitative comparison between baseline and our proposed approach. `VoRTeC` (Ours) is capable of reconstructing complex textures with realism, even at extremely low bit rates. *Best viewed when zoomed in.*

standards in rate–distortion optimization, yet distortion-driven objectives (e.g., MSE) inherently oversmooth textures and erase fine structures when the bitrate drops to the ultra-low regime, causing a sharp decline in perceptual realism.

To improve perceptual quality at low bitrates, a growing body of work introduces generative priors into the compression pipeline—a paradigm known as generative compression [Mentzer et al., 2020, Careil et al., 2024, Vonderfecht and Liu, 2025]. In the image domain, large-scale GANs [Goodfellow et al., 2014] and diffusion models [Ho et al., 2020, Rombach et al., 2022] have been successfully leveraged to recover high-frequency details, yielding visually convincing reconstructions even below 0.01 bpp. Extending this paradigm to video, however, introduces a fundamentally stricter requirement: *temporal coherence*. Early perceptual video codecs [Mentzer et al., 2022, Yang et al., 2022] incorporate adversarial training or perceptual losses, but their limited generation capacity still yields noticeable artifacts. More recent diffusion-based video codecs [Ma and Chen, 2025, Qi et al., 2025] adopt pre-trained *image* diffusion priors for frame-wise enhancement. While each frame may appear sharp in isolation, image-domain priors lack any modeling of temporal dynamics, causing restored textures to drift across frames—a phenomenon known as *perceptual flickering*—which becomes especially severe at ultra-low bitrates.

Video-native diffusion models, particularly diffusion transformers (DiTs) [Peebles and Xie, 2023, Wang et al., 2025, Yang et al., 2024], offer a natural remedy. Trained on large-scale video data, they learn spatio-temporal latent representations that capture appearance, motion, and long-range dependencies within a unified structure. Very recently, GNVC-VD [Mao et al., 2025] demonstrated that integrating a pre-trained video diffusion transformer (Wan2.1 [Wang et al., 2025]) into neural video compression can produce coherent fine textures with strong temporal stability. However, this benefit comes at a prohibitive cost: the decoder must perform *multi-step* flow-matching inference—typically 5 or more sequential denoising iterations through a multi-billion-parameter DiT. Similarly, Free-GVC [Ling et al., 2026] reformulates compression as progressive coding along the diffusion trajectory, but its multi-step reverse process yields less than 0.5 fps even at 480p. In essence, existing video-native generative codecs face a fundamental *speed–quality dilemma*: the multi-step sampling process that endows video diffusion priors with their generative power is precisely what makes them impractical for real-time decoding.

A parallel line of research addresses diffusion efficiency through distillation. Distribution Matching Distillation (DMD) [Yin et al., 2024a,b] and consistency models [Song et al., 2023] enable single-step generation in the image domain, and S2VC [Xue et al., 2025a] has recently applied single-step image diffusion to video compression. Yet these methods remain anchored to *image* diffusion backbones, which lack the spatio-temporal modeling required for coherent video reconstruction. The key question is therefore: *can we fully use the rich spatio-temporal prior of a video-native flow-matching foundation model into a single decoding step, while preserving both perceptual fidelity and inter-frame consistency?*

We propose VoRTeC, a **R**eal **T**ime **V**ideo **C**ompr**e**ssion framework built upon a **o**ne-step foundational Flow model, as one solution to this problem. VoRTeC compresses spatiotemporal latents into compact transmittable representations, and treats the decoding result as a temporal state along the flow path. This enables us to perform enhanced decoding even without access to the parameters or gradients of the foundation flow-matching network. Then, through a ViT-based prior fusion network, VoRTeC fuses and aligns the coarse-grained compressed representation with the fine-grained prior representation, ultimately achieving fast single-step decoding. Furthermore, we design an inter-group communication technique (CGG), which allows spatiotemporal information from preceding video groups to flow into subsequent groups. This ensures inter-group consistency and further reduces the bitrate without incurring additional training or decoding overhead. We conduct extensive experiments at various resolutions to validate the efficiency of VoRTeC: VoRTeC achieves a decoding speed of *3.93 fps on 1080p* datasets and enables **real-time** decoding at *32 fps on 480p*. Compared with previous diffusion-based baselines, our method achieves drastically reduced decoding latency while improving perceptual quality by approximately 58.12%–73.25%, as illustrated in Figure 1. Furthermore, due to its externally mounted architectural design, VoRTeC exhibits low training complexity; consequently, even when employing Wan2.1 [Wan et al., 2025] as the backbone, it can be effectively trained on a single NVIDIA A6000 GPU.

Building upon this, we further propose VoRTeC$^+$, which fine-tunes the flow-matching network with LoRA on top of VoRTeC. Benefiting from the excellent prior pattern learning paradigm of VoRTeC, VoRTeC$^+$ improves perceptual performance by 20–30% over the original scheme, while introducing only 10M additional parameters and leaving the inference speed unaffected, surpassing all existing video compression approaches built upon foundation flow models. As shown in the Figure 1, our proposed method can reconstruct finer and more realistic details, and significantly outperforms the prior generative method GLC-Video [Qi et al., 2025] in terms of temporal consistency.

In summary, our contributions can be summarized as follows:

- We propose a latent codec that re-compresses spatiotemporal latent representations into compact, quantization-ready codes and employ it to initialize the flow-matching process by mapping these compressed representations onto the corresponding flow trajectories. This strategy enables the framework to capture the underlying prior distribution even when the initial gradients of the flow-matching network are unavailable.
- We design the Flow Prior Multi-Fusion module to perform multi-scale fusion and alignment between the prior and compressed latent. On one hand, this enables the compressor to better remove prior redundancy; on the other hand, it further refines the prior representation. Meanwhile, we propose an inter-group communication algorithm that ensures temporal consistency across frame groups in a training-free manner.
- Building upon these insights, we propose VoRTeC and VoRTeC$^+$. Through solely external module and algorithm design, VoRTeC learns the prior distribution of the foundation flow and achieves high-fidelity single-step decoding. Compared to diffusion-based video compression baselines, VoRTeC achieves BD-rate savings of 58.12%–73.25% in terms of LPIPS. Through further LoRA fine-tuning, VoRTeC$^+$ extends this performance advantage to 64.53%–74.8%. Compared to previous Wan2.1 based method Mao et al. [2025], VoRTeC$^+$ achieves a 23.44% improvement in perceptual quality (DISTS). Meanwhile, VoRTeC substantially improves the decoding speed over previous diffusion-based methods ($3 \sim 197\times$), achieving a decoding frame rate of 32 fps on 480p and requiring only 0.25 s to decode a single 1080p frame.

2 Related Work

2.1 Neural Video Compression

Neural video compression (NVC) has advanced rapidly, leveraging end-to-end learning to surpass traditional hybrid codecs in performance. The fundamental objective of lossy compression is the rate-distortion optimization:

$$\min_{\boldsymbol{\theta}} \; R(\hat{\mathbf{x}};\, \boldsymbol{\theta}) + \lambda\, D(\mathbf{x},\, \hat{\mathbf{x}})\,, \tag{1}$$

where $R(\hat{\mathbf{x}}; \boldsymbol{\theta})$ estimates the bitrate of the compressed representation $\hat{\mathbf{x}}$, $D(\cdot,\cdot)$ is a distortion metric (e.g., MSE), and λ controls the rate–distortion trade-off. DVC [Lu et al., 2019] introduced the first end-to-end framework that jointly optimizes motion estimation, residual coding, and entropy modeling. DCVC [Li et al., 2021] shifted the paradigm from residual coding to conditional coding, allowing the model to learn more informative temporal context from decoded features. Subsequent extensions progressively enhanced context modeling: DCVC-FM [Li et al., 2024] incorporated feature modulation for finer rate control, and DCVC-RT [Jia et al., 2025] optimized both architecture and runtime efficiency to achieve real-time decoding on consumer hardware. Other lines of work explore hierarchical variational autoencoders [Lu et al., 2024] and non-local temporal correlations [Jiang et al., 2025]. Despite these advances, distortion-oriented NVCs optimized for MSE or MS-SSIM [Wang et al., 2004] inevitably produce over-smoothed reconstructions at ultra-low bitrates, lacking fine textures and perceptual realism.

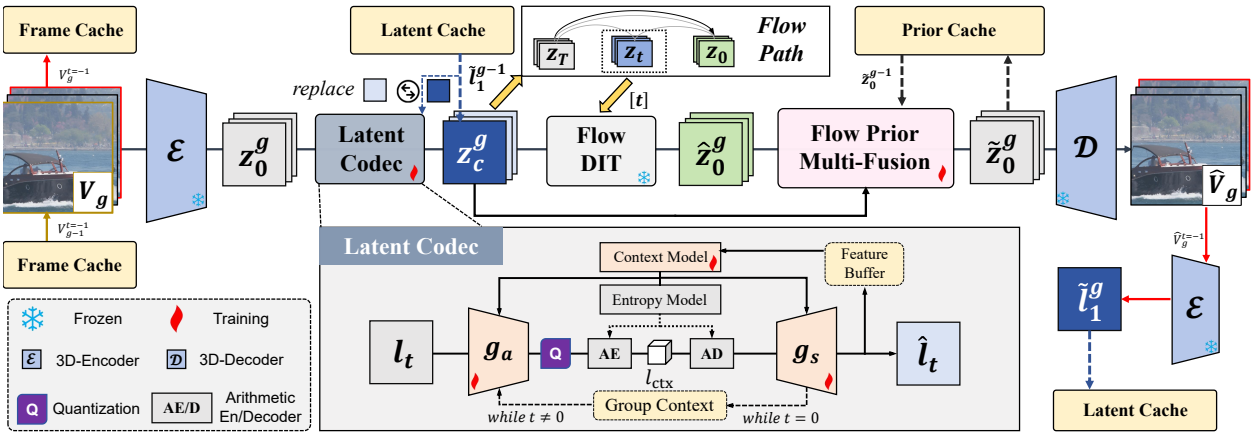


Figure 2: Overall pipeline of VoRTeC. $\mathcal{E}$ and $\mathcal{D}$ denote the pre-trained 3D video en/decoder [Wang et al., 2025], and FlowDIT adopts the flow-matching network from Wan2.1 (1.3B) Wang et al. [2025].

2.2 Generative Compression

Generative compression leverages learned generative models to recover realistic textures beyond pixel-level fidelity, making it particularly effective at ultra-low bitrates. The shift from classical R–D optimization to the rate–distortion–perception (R–D–P) trade-off [Blau and Michaeli, 2019] can be formalized as:

$$\min_{\boldsymbol{\theta}} \; R(\hat{\mathbf{x}};\, \boldsymbol{\theta}) + \lambda\, D(\mathbf{x},\, \hat{\mathbf{x}}) + \gamma\, d(p_{\mathbf{x}},\, p_{\hat{\mathbf{x}}})\,, \tag{2}$$

where $d(\cdot,\cdot)$ measures the distributional discrepancy between the real and reconstructed data distributions $p_{\mathbf{x}}$ and $p_{\hat{\mathbf{x}}}$, and γ controls the emphasis on perceptual realism.

In the image domain, HiFiC [Mentzer et al., 2020] pioneered perceptual and adversarial losses with a generative decoder, while subsequent work [Li et al., 2025, Xia et al., 2025, 2026, Jia et al., 2026] increasingly turned to diffusion models as stronger priors that guide multi-step denoising with compact semantic latents. To address sampling latency, StableCodec [Zhang et al., 2025] and OneDC [Xue et al., 2025b] distill the diffusion process into a single step, achieving over 20× speedup. These advances, however, are inherently limited to single-frame reconstruction without temporal modeling. Extending generative compression to video, GLC-Video [Qi et al., 2025] and DiffVC [Ma and Chen, 2025] adopt image-domain priors for frame-wise enhancement but inevitably suffer from temporal flickering due to the absence of inter-frame modeling. GNVC-VD [Mao et al., 2025] introduced the first video-native diffusion prior, achieving temporally coherent refinement via multi-step flow matching, while S2VC [Xue et al., 2025a] pursued faster single-step image-diffusion decoding at the cost of inter-frame consistency.

2.3 Video Diffusion Models and Flow Matching

Diffusion models [Ho et al., 2020] have advanced video generation from 2D U-Nets [Ho et al., 2022] through latent diffusion [Rombach et al., 2022] and video latent models [Yang et al., 2024, Blattmann et al., 2023] to the diffusion transformer (DiT) [Peebles and Xie, 2023], which models videos as token sequences for long-range temporal reasoning. Wan2.1 [Wang et al., 2025] further advanced this paradigm with a spatio-temporal VAE and a flow-matching [Lipman et al., 2023, Liu et al., 2023] formulation. Unlike DDPM's discrete Markov chain, flow matching learns a continuous velocity field trained via:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,\,\mathbf{x}_0,\,\mathbf{x}_1}\left[\left\|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1 - \mathbf{x}_0)\right\|^2\right], \tag{3}$$

where $\mathbf{x}_t = (1-t)\,\mathbf{x}_0 + t\,\mathbf{x}_1$, $t \in [0,1]$, and $\mathbf{v}_\theta$ is a DiT-based velocity network.

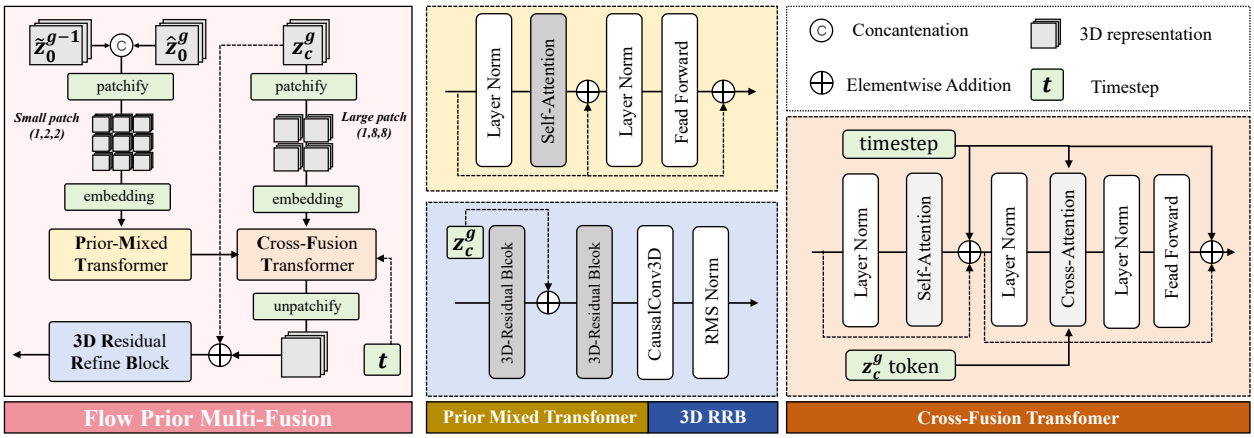


Figure 3: Architecture of the Flow Prior Multi-Fusion (FPMF) module. By adopting patchify strategies with different granularities, FPMF achieves precise alignment between prior texture and structural features of compressed latent, facilitating more compact encoding and high-fidelity reconstruction.

3 Methodology

3.1 Framework Overview

The framework of our proposed `VoRTeC` is illustrated in Figure 2: given an input group of frames $\boldsymbol{V}_g \in \mathbb{R}^{(1+T)\times H\times W\times 3}$, a 3D analysis encoder $\mathcal{E}(\cdot)$ compresses it into a compact spatiotemporal latent representation:

$$\{l_t^g\}_{t=1}^{1+T/4} = z_0^g = \mathcal{E}(\boldsymbol{V}_g) \tag{4}$$

$\mathcal{E}(\cdot)$ performs temporal modeling through causal convolution: it encodes the first frame into an Intra-latent (I-latent) l_0^g, and applies temporal downsampling to the subsequent frames, encoding them as Predictive latents (P-latents) $l_{t\geq 1}^g$. This ensures spatiotemporal consistency within each frame group, but continuity across frame groups cannot be guaranteed. Inspired by [Sullivan et al., 2012], we partition the entire video sequence into key groups (also referred to as I-Groups) and predictive groups (P-Groups).

3.1.1 Key Group Compression

We refer to the first frame group of a meta-group as the key group. To reduce intra-group redundancy, our latent codec needs to model latent-wise context. Specifically:

$$\hat{y}_t^g = \mathbf{Q}\left(\boldsymbol{g}_a\left(l_t^g \mid \hat{l}_{t-1}^g, l_{ctx}\right)\right), \hat{l}_t^g = \boldsymbol{g}_s\left(\hat{y}_t^g, \hat{l}_{t-1}^g, l_{ctx}\right). \tag{5}$$

Here, l_{ctx} denotes the group context, which is extracted from the I-latent l_1^g. Since the causal encoder concentrates spatial information into l_1^g, during compression we not only use the cached previous latent but also employ the I-latent as the group-wide context for compression.

Subsequently, we estimate $z_c^g = \{\hat{l}_t^g\}_{t=1}^{1+T/4}$ to determine the approximate position of the decoded compressed representation along the flow path:

$$\lfloor t \rceil = \Gamma(z_c^g, z_0^g). \tag{6}$$

This information is encoded and transmitted to the decoder so that FlowDIT can treat the compressed representation as a state along the flow and perform flow-matching denoising:

$$\hat{z}_0^g = \mathbf{FlowDIT}(z_c^g, \lfloor t \rceil) \tag{7}$$

For further refinement, the flow-prior reconstruction $\hat{z}_0^g$ is fed into the prior fusion network $\mathcal{F}(\cdot,\cdot,\cdot)$ together with the compressed representation z_c^g:

$$\tilde{z}_0^g = \mathcal{F}(\hat{z}_0^g, z_c^g, z_{ctx}) \tag{8}$$

Here, z_{ctx} is the cached prior context. In the key group, since no cache is available, we use z_c^g as a substitute.

Through end-to-end training, this network implicitly fuses and aligns the flow representation domain and the compressed representation domain. Subsequently, $\tilde{z}_0^g$ is stored in the prior cache, serving as z_{ctx} to assist reconstruction in P-groups.

Finally, at the decoder side, the 3D analysis decoder $\mathcal{D}(\cdot)$ decodes the reconstructed representation to the image domain:

$$\hat{\boldsymbol{V}}_g = \mathcal{D}(\tilde{z}_0^g)$$

3.1.2 Contact Group to Group (CGG)

Pre-trained video foundation flows cannot guarantee consistency across frame groups, and blindly increasing the number of frames within a group not only incurs substantial computational overhead but also limits the algorithm's adaptability to videos of different lengths. To address this, we design a non-intrusive caching mechanism to establish connections between groups, as shown in the Figure 2: the last frame of the I-Group enters the frame cache to serve as the first frame of the P-Group; simultaneously, the last frame of the reconstructed video $\hat{\boldsymbol{V}}_g$ from the I-Group is re-encoded into the latent cache for subsequent use.

Subsequently, when the compressor performs latent compression via (2), we use the cached $\tilde{l}_1^{g-1}$ from the latent cache to extract the group context l_{ctx}, and reconstruct the compressor's decoding output z_c^g:

$$z_c^g = \{\tilde{l}_1^{g-1}\} + \{\hat{l}_t^g\}_{t=2}^{1+T/4}$$

In this way, we encode the temporal context into the latent representation through the analysis encoder $\mathcal{E}(\cdot)$, but we neither encode nor transmit the key latent; instead, we substitute it with the cached latent. Grounded on the last decoded frame of the preceding group, this inter-frame spatiotemporal information guides the reconstruction of subsequent frames, thereby maintaining inter-group consistency. Meanwhile, the prior reconstruction result of the preceding group also enters the prior cache, to be retrieved as z_{ctx} during prior fusion in subsequent groups.

This design offers three advantages. First, it fully exploits inter-group context, further reducing the bitrate while maintaining reconstruction quality. Second, it avoids the artifacts—such as flickering and visual discontinuities—that would arise from inter-group independence in the reconstructed video. Third, it allows flexible adjustment of the ratio between P-Groups and I-Groups, enabling training-free bitrate selection within a certain range.

3.2 Compress latent for flow-matching

We aim to treat the decoded compressed representation as a state along the flow path, thereby non-intrusively learning the prior distribution pattern of the foundation flow-matching model. To this end, we need to compute the shortest distance from z_c^g to the flow path. We assume that the compressed representation follows a normal distribution: $z_c^g \sim \mathcal{N}\left(z_0^g, \sigma^2\right)$. Given time t, the distribution along the flow path is $\mathbf{FL}(t) \sim \mathcal{N}\left((1-t)z_0^g, t^2\right)$. We seek to find the following time step $t^\star$:

$$t^\star = \arg\min_t \mathbf{dist}\left[(1-t) * z_c^g, \mathbf{FL}(t)\right], t \in [0, 1]$$

Here, we also apply a corresponding scaling to z_c^g to accommodate the mean shift caused during the flow matching process. $\mathbf{dist}$ denotes a certain distance metric. When we use the Wasserstein distance, the solution can be readily obtained (see the appendix for the proof):

$$t^\star = \frac{\sigma}{1+\sigma}$$

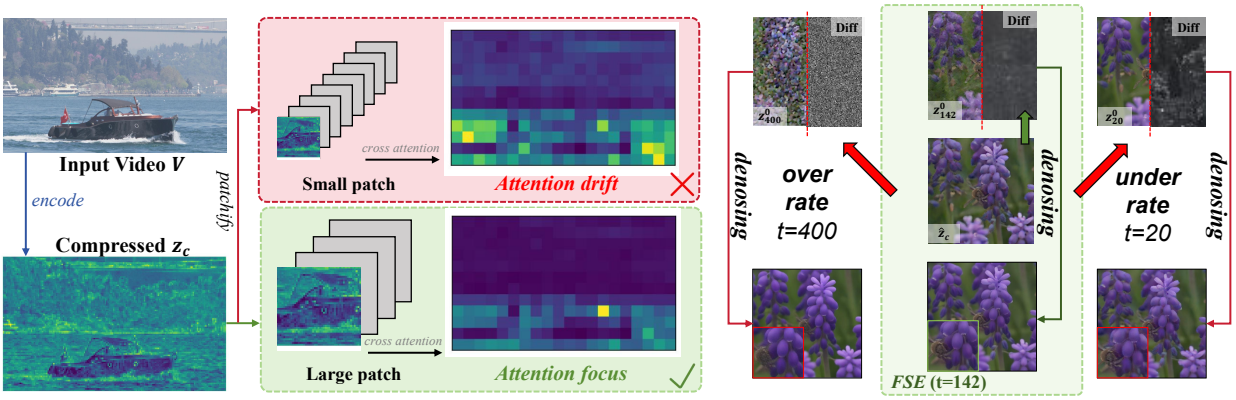
At this point, we can estimate the variance $\sigma^2 = \|z_c^g - z_0^g\|_2$ based on the compressed representation z_c^g and the input latent z_0^g, compute $t^\star$, and round it to obtain $\lfloor t \rceil = \mathbf{round}(t * 1000)$. For video transmission tasks, the number of bits required to transmit the integer $\lfloor t \rceil$ is minimal and almost negligible.

After estimation, we treat z_c^g as the flow state at time step $\lfloor t \rceil$, and use the flow-matching network for denoising:

$$\hat{z}_0^g = \mathbf{FlowDIT}(z_c^g, \lfloor t \rceil) \tag{9}$$

$$= (1 - \frac{\lfloor t \rceil}{1000}) z_c^g - \mathbf{v}_\theta((1 - \frac{\lfloor t \rceil}{1000}) z_c^g, \lfloor t \rceil, ctx) \tag{10}$$

Here, ctx is a fixed text semantic embedding. As shown in the Figure 4(b), our proposed estimation method avoids the absence of denoising and the over-smoothing caused by erroneous temporal estimation, thereby better learning the distribution pattern of the flow prior.



(a) Attention map visualization in the FPMF Cross Fusion Transformer.

(b) Visualization results of distinct temporal flow states.

Figure 4: Visualizations of the modules and selected methods. (a) Coarse-grained patchify drives compressed tokens to capture structural information, while fine-grained patchify distributes attention over latent space. (b) Overestimated or underestimated predictions cause the denoising network to either skip denoising entirely or render the image excessively smooth. Flow-State Estimation (FSE) precisely calculates the temporal state with high accuracy.

3.3 Fusing Flow Prior for E2E training

The prediction of the flow-matching network cannot be directly used as the reconstruction result, for two main reasons. First, the error in time estimation and the discrepancy between the compressed representation distribution and the true flow path distribution would degrade reconstruction quality. Second, without accessing the gradients of the flow-matching network, the compressor cannot remove prior redundancy through end-to-end training. To address this, we design **F**low **P**rior **M**ulti **F**usion (FPMF), a Transformer-based prior correction and alignment network, whose architecture is shown in the Figure 3.

The network takes as input the flow-prior reconstruction of the current latent group $\hat{z}_0^g$, the prior reconstruction of the preceding group $\hat{z}_0^{g-1}$, and the compressed representation z_c^g. The prior reconstructions are first concatenated and patchified into fine-grained small patches, since compared to the compressed representation, the prior reconstruction latents contain more high-frequency details, and we need a smaller patch size to attend to these textures as much as possible. In contrast, due to the ill-posed degradation and distortion introduced by compression, the details in z_c^g are lost and unreliable. Therefore, we adopt a coarse-grained patchify strategy to learn the low-frequency non-local features of the compressed representation. The prior patches then pass through stacked ViT layers for mixing, and are subsequently fused and matched with z_c^g in the Cross Fusion Transformer. As shown in Figure 4 (a), when $\hat{z}_0^{g-1}$ is patchified into fine-grained patches, the attention is dispersed across texture-rich regions, which is unhelpful for reconstruction. Coarse-grained patches, on the other hand, avoid this issue and concentrate the attention on semantically rich regions such as contours and objects.

In fact, FPMF not only improves reconstruction performance but also further reduces the bitrate through a modeling approach guided by the principle of "compressing low-frequency structures while restoring high-frequency details via priors." We also verify this in the ablation experiments.

3.3.1 Training Strategy

`VoRTeC` is optimized end-to-end through rate loss and reconstruction loss across all modules. The rate loss constrains the bitrate of the compressor:

$$\mathcal{L}_{rate} = R_{z_0^g} + \mathcal{L}_{\text{codebook}}$$

Here, the rate $R_{z_0^g}$ is the aggregate of the rates of all frames: $R_{z_0^g} = \sum_{t=1}^{1+T/4} R_{l_t^g}$.

The reconstruction loss can be divided into three parts. The first part is the reconstruction loss in the latent domain $\mathcal{L}_{\textbf{latent}}$, which constrains the compressor and the FPMF module:

$$\mathcal{L}_{\textbf{latent}} = \|z_0^g - z_c^g\|_2 + \|\hat{z}_0^g - z_0^g\|_2$$

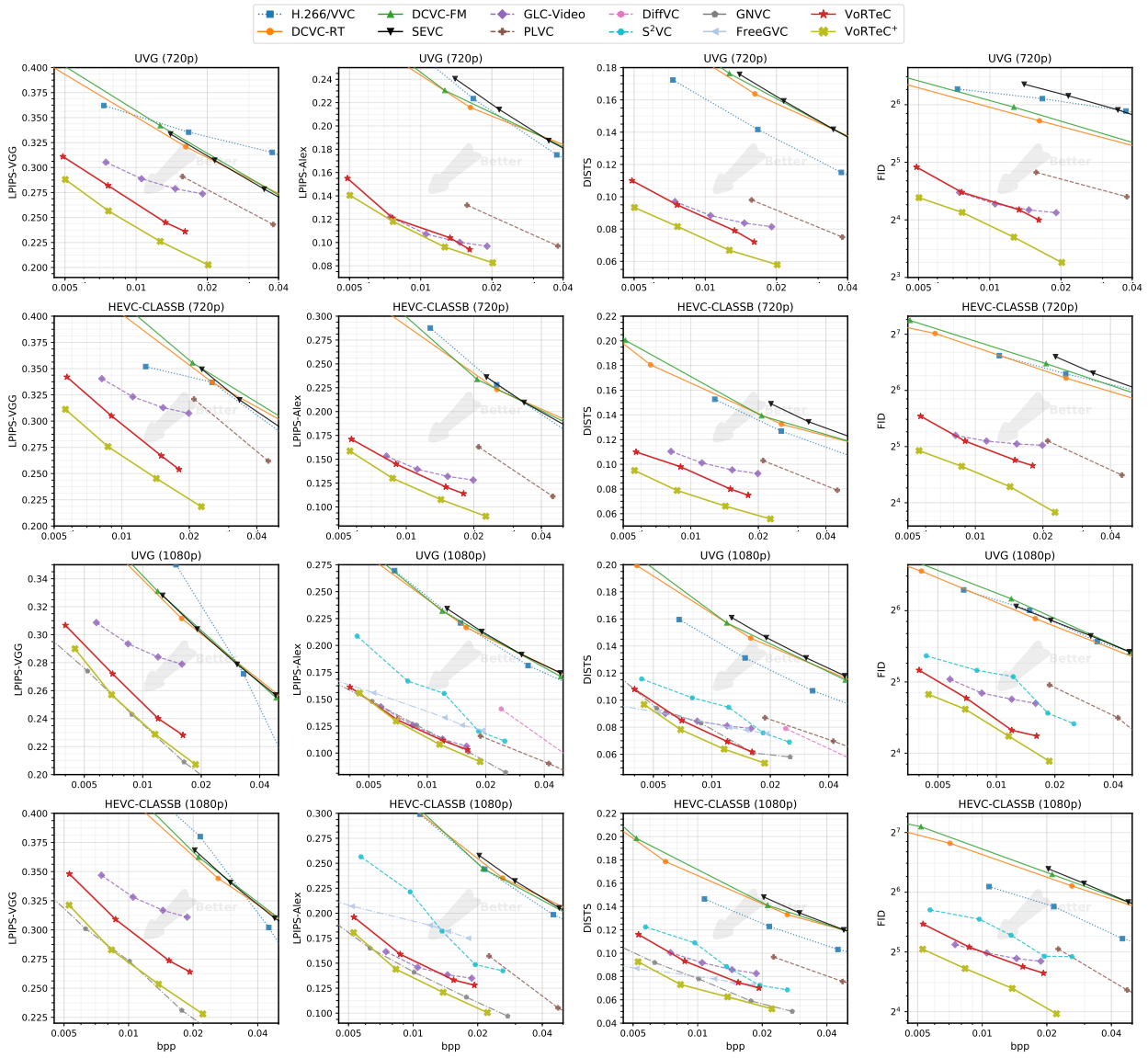


Figure 5: Rate–distortion/perception curves on HEVC Class B and UVG datasets at 720p and 1080p resolutions.

The second part is the reconstruction loss in the image domain $\mathcal{L}_{\mathbf{rec}}$, which further improves the accuracy and perceptual quality of the reconstruction:

$$\mathcal{L}_{\mathbf{rec}} = \|\hat{\boldsymbol{V}}_g - \boldsymbol{V}_g\|_2 + \gamma_1 \mathcal{L}_{\mathbf{lpips}}(\hat{\boldsymbol{V}}_g, \boldsymbol{V}_g) + \gamma_2 \mathcal{L}_{\mathbf{dino}}(\hat{\boldsymbol{V}}_g, \boldsymbol{V}_g)$$

The third part is the adversarial loss $\mathcal{L}_{\mathbf{adv}}(\hat{\boldsymbol{V}}_g, \boldsymbol{V}_g)$, which narrows the gap between the reconstructed video distribution and the real video distribution. In summary, the overall training loss is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{rate} + \lambda_2(\mathcal{L}_{\mathbf{latent}} + \mathcal{L}_{\mathbf{rec}}) + \gamma_3 \mathcal{L}_{\mathbf{adv}}$$

Here, λ_1 and λ_2 are hyperparameters that control the rate-distortion trade-off, and γ_1, γ_2, γ_3 are fixed hyperparameters.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our method on three standard video compression benchmarks: UVG [Mercat et al., 2020], HEVC Class B,C [Sullivan et al., 2012], and MCL-JCV [Wang et al., 2016]. UVG, HEVC Class B and MCL-JCV

contain 1920×1080 high-resolution natural videos. HEVC Class C provides challenging test sequences at 832×480 resolution. Following common practice in learned video compression [Li et al., 2023, Jia et al., 2025], we test HEVC Class B and UVG under two protocols: (i) center-cropping to 1280×720 and encoding the full sequence; and (ii) evaluating the first 96 frames at the native 1920×1080 resolution. All frames are processed in the RGB domain. For results on MCL-JCV and HEVC-Class C, please refer to the appendix.

Baselines. We compare against a comprehensive set of traditional, neural, and generative video codecs. For traditional standards, we use VTM-17.0 [Bross et al., 2021], the reference implementation of H.266/VVCs. For neural video codecs, we include DCVC-RT [Jia et al., 2025], DCVC-FM [Li et al., 2024], and SEVC [Bian et al., 2025], representing the recent advances in conditional coding and spatially embedded compression. For generative compression, we compare with GLC-Video [Qi et al., 2025], PLVC [Yang et al., 2022], DiffVC [Ma and Chen, 2025], S2VC [Xue et al., 2025a], covering GAN-based and diffusion-based approaches. For certain diffusion-based baseline methods [Mao et al., 2025, Li et al., 2026] for which the source code is not publicly available, we report the performance metrics as presented in the respective original publications.

Metrics. We employ a suite of widely recognized quantitative metrics to assess the visual quality, including perceptual metrics such as LPIPS [Zhang et al., 2018] (vgg as default), DISTS [Ding et al., 2020] and FID [Heusel et al., 2017]. Further, we utilize distortion metrics like PSNR. We additionally report FloLPIPS [Danier et al., 2022] and E_{warp} [Lai et al., 2018] to quantify temporal flickering artifacts. Meanwhile, we also report the BD-Rate [Bjontegaard, 2001] as the metric for rate-distortion performance.

4.2 Main Results

Figure 5 compares `VoRTeC` against baselines on HEVC-B and UVG. Traditional and distortion-optimized neural encoders (H.266/VVC, DCVC-RT, DCVC-FM Li et al. [2024], SEVC) suffer from over-smoothing below 0.03 bpp, reflected by high perceptual metric values. Generative baselines (GLC-Video, PLVC, DiffVC, S2VC) improve perceptual quality via adversarial or diffusion priors, yet remain inferior to `VoRTeC`. At 720p, `VoRTeC` achieves consistent perceptual superiority without sacrificing distortion—it also outperforms the previous SOTA generative scheme GLC-Video in PSNR. At 1080p the advantage narrows but stays leading, which we attribute to the Wan2.1 1.3B base model being trained on 480p generation. `VoRTeC` shows consistent advantages on HEVC-ClassC (480p) as reported in the appendix, and qualitative visual comparisons in the supplementary materials further confirm its benefits. Benefiting from LoRA fine-tuning, `VoRTeC`$^+$ enables the prior distribution to adapt to high-resolution compression degradation, yielding considerable performance gains. It substantially surpasses nearly all other generative video compression baselines in perceptual metrics and exhibits consistent superiority over GLC-Video. We further present quantitative experiments on inter-frame continuity of decoded frames in Figure 6. Our method outperforms other learning-based video compression approaches across both metrics. Compared with GNVC [Mao et al., 2025], which also adopts Wan2.1 as the backbone, `VoRTeC` achieves stronger inter-frame consistency, which demonstrates the contribution of our scheme to extending the long-range context of the model.

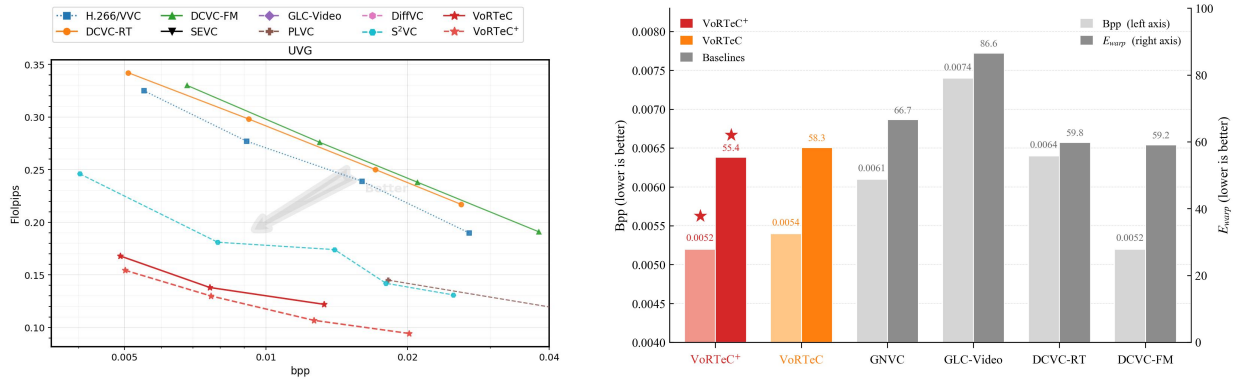
Complexity analysis. Table 1 reports encoding and decoding throughput in frames per second (fps) across four resolutions on a single NVIDIA A100 GPU. VTM17 was tested on an Intel Xeon Gold 6330 CPU. As can be observed, `VoRTeC` effectively addresses the speed-quality dilemma. Although `VoRTeC` builds upon a billion-parameter video flow-matching foundation model, it achieves a decoding speed of 3.93 frames per second at 1080p resolution. Specifically, it is $6.1\times$ faster than GNVC-VD [Mao et al., 2025] (which also adopts Wan2.1), $4.1\times$ faster than YODA [Li et al., 2026], and $3.1\times$ faster than S2VC, while maintaining comparable speed with DCVC-DC [Li et al., 2023] and DCVC-FM. The speed advantage of `VoRTeC` becomes more pronounced at lower resolutions. It achieves decoding frame rates of 105.25, 32.31, and 12.60 fps at resolutions of 240p, 480p, 720p respectively. These results outperform all learned baselines, including DCVC-FM and GLC-Video.

4.3 Qualitative Results

We present visual results of various state-of-the-art methods on the HEVC and UVG datasets, as shown in fig. 7. Compared to MSE-optimized codecs (SEVC, DCVC series), `VoRTeC` reconstructs more realistic video frames, with clearer facial contours and details. After LoRA fine-tuning, `VoRTeC`$^+$ achieves improved reconstruction of details and textures, accompanied by a moderate reduction in bitrate.

4.4 Ablation Studies

Module Ablations Table 2 (top) presents ablation experiments for individual modules.



(a) Rate–flolpips curves on UVG datasets at 1080p resolutions. (b) Comparison of inter-frame continuity for different methods.

Figure 6: Quantitative experiments for verifying inter-frame continuity/consistency. (a) FloLPIPS metrics of various methods tested on the UVG dataset. Our proposed method consistently outperforms the baselines. (b) E_{warp} metrics and corresponding bpp values of various methods tested on the HEVC-ClassB dataset. `VoRTeC` achieves the minimal E_{warp} even at the lowest bpp, indicating superior inter-frame continuity.

(a) Original (b) VTM-17.0 (c) DCVC-RT (d) GLC-Video (e)VoRTeC (d)VoRTeC+

Figure 7: Qualitative illustrations of various methods. *Best viewed when zoomed in.*

Flow State Estimation: Instead of estimating the temporal state of the compressed representation, we use a fixed timestamp (t=300) during training. As can be seen, the perceptual quality degrades noticeably, indicating that the model struggles to learn the prior distribution pattern.

Prior Fusion: We directly remove the FPMF module with no refinement on flow-prior reconstruction. As prior gradient information is unavailable in training, prior redundancy cannot be easily eliminated from the compressed representation, leading to a notable drop in perceptual quality.

Multi-patch: For both the compressed representation and the prior reconstruction, we use the same mini patch size. As shown in Figure 4, this causes attention dispersion and leads to a certain degree of degradation in reconstruction performance.

Prior Cache: We do not cache the prior representation $\tilde{z}_0^g$; instead, FPMF uniformly uses z_c^g as the auxiliary input. This weakens the information flow between groups, leading to a certain degree of inter-frame instability and perceptual quality degradation.

Lora Fintune: We fine-tune FlowDiT using LoRA, namely `VoRTeC`$^+$ mentioned in the main method. Fine-tuning enables the denoising network to adapt to compression noise, which yields a noticeable improvement in decoding perceptual performance. Meanwhile, our scheme implicitly aligns compression noise with Gaussian noise; therefore, simple fine-tuning allows the compression framework to achieve SOTA performance.

Table 1: Encoding and decoding throughput (fps) across four resolutions on a single A800 GPU. Higher is better.

Resolution	416×240		832×480		1280×720		1920×1080	
Method	Enc. Fps	Dec. Fps	Enc. Fps	Dec. Fps	Enc. Fps	Dec. Fps	Enc. Fps	Dec. Fps
VTM-17	0.47	121.73	0.12	56.62	0.08	36.92	0.02	12.35
DCVC-DC	36.24	43.88	15.63	18.71	7.47	9.36	3.43	4.36
DCVC-FM	36.51	43.09	16.82	17.85	7.58	8.94	3.51	4.22
GLC-Video	49.39	57.22	28.60	19.92	13.25	8.82	6.36	4.19
DiffVC	0.85	0.88	0.29	0.29	0.09	0.09	0.02	0.02
GNVC-VD	—	—	< 40.00	< 7.75	17.24	2.59	6.54	0.64
FreeGVC	—	—	$0.25 \sim 1.38$	$0.32 \sim 1.25$	—	—	—	—
S2VC	—	—	—	—	—	—	6.60	1.27
YODA	—	—	—	—	—	—	—	0.97
`VoRTeC` (Ours)	84.46	105.25	20.34	32.31	8.76	12.60	3.84	3.93

Module Ablation			Loss Ablation		
	BD-rate%(LPIPS)	BD-rate%(FID)		BD-rate%(LPIPS)	BD-rate%(FID)
W/o FSE	44.31	38.52	W/o $\mathcal{L}_{\mathbf{lpips}}$	40.31	28.32
W/o FPMF	72.59	91.06	W/o $\mathcal{L}_{\mathbf{dino}}$	12.64	21.37
W/o Multi-patch	16.24	22.7	W/o $\mathcal{L}_{\mathbf{adv}}$	18.02	37.15
W/o Prior-cache	19.17	23.83			
W/ Lora Finetune	-30.44	-45.32			

Table 2: Ablation study on module components (top) and loss functions (bottom). BD-rate (%) is computed relative to the full `VoRTeC` model on the UVG dataset. Lower is better.

Loss Ablations Table 2 (bottom) summarizes ablation experiments for different reconstruction loss functions. Both $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{dino}}$ effectively boost the visual realism of reconstructed images. Moreover, $\mathcal{L}_{\text{adv}}$ yields more prominent improvements in depth estimation accuracy for decoded frames. These observations reveal that enforcing feature-level consistency between reconstructed and original images within self-supervised model space helps preserve semantic and depth information of compressed images to the greatest extent.

5 Conclusion

This work presents `VoRTeC`, a novel compression framework that enables one-step decoding using the pre-trained Wan2.1 video flow-matching foundation model. `VoRTeC` non-intrusively embeds spatio-temporal representations into the flow path and refines latent features via a multi-scale prior fusion module. Meanwhile, the proposed CGG scheme facilitates information interaction among frame groups and improves temporal continuity with negligible overhead. Extensive experiments show that `VoRTeC` achieves an excellent rate-perception tradeoff and substantially accelerates decoding compared with previous diffusion-based methods, enabling real-time 32 FPS decoding at 480p. Benefiting from its flexible design, `VoRTeC` is compatible with flow-matching foundation models of various scales and architectures, providing a new paradigm for generative video compression.

References

Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

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.

Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. In *Advances in Neural Information Processing Systems*, volume 34, pages 18114–18125, 2021.

Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22616–22626, 2023.

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.

Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Libiao 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.

Fabian Mentzer, George Toderici, Michael Tschannen, Eirikur Agustsson, and Mario Lucic. High-fidelity generative image compression. In *Advances in Neural Information Processing Systems*, volume 33, pages 14013–14024, 2020.

Maximilian 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.

Joel Vonderfecht and Feng Liu. Lossy compression with pretrained diffusion models. In *International Conference on Learning Representations*, 2025.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

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.

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

Ren Yang, Radu Timofte, and Luc Van Gool. Perceptual learned video compression with recurrent conditional GAN. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1537–1544, 2022.

Wei Ma and Zhibo Chen. Diffusion-based perceptual neural video compression with temporal diffusion information reuse. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 21(12):1–22, 2025.

Libiao 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.

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.

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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 annotated transformer. *arXiv preprint arXiv:2408.06072*, 2024.

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.

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.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. One-step diffusion with distribution matching distillation. In *International Conference on Machine Learning*. PMLR, 2024a.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32111–32152. PMLR, 2023.

Naifu Xue, Zhaoyang Jia, Jiahao Li, Bin Li, Zihan Zheng, Yuan Zhang, and Yan Lu. Single-step diffusion-based video coding with semantic-temporal guidance. *arXiv preprint arXiv:2512.07480*, 2025a.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Ming Lu, Zhuoyuan Duan, Feng Zhu, and Zhan Ma. Deep hierarchical video compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8859–8867, 2024.

Wei Jiang, Jiahao Li, Kai Zhang, and Li Zhang. ECVC: Exploiting non-local correlations in multiple frames for contextual video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2025.

Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. *IEEE Transactions on Image Processing*, 13(4):668–681, 2004.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pages 675–685. PMLR, 2019.

Zhiyuan Li, Yanhui Zhou, Hao Wei, Chenyang Ge, and Jingwen Jiang. Towards extreme image compression with latent feature guidance and diffusion prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(1): 888–899, 2025.

Yichong Xia, Yimin Zhou, Jinpeng Wang, Baoyi An, Haoqian Wang, Yaowei Wang, and Bin Chen. DiffPC: Diffusion-based high perceptual fidelity image compression with semantic refinement. In *International Conference on Learning Representations*, 2025.

Yichong Xia, Yimin Zhou, Jinpeng Wang, and Bin Chen. Towards efficient low-rate image compression with frequency-aware diffusion prior refinement. *arXiv preprint arXiv:2601.10373*, 2026.

Zhaoyang Jia, Naifu Xue, Zihan Zheng, Jiahao Li, Bin Li, Xiaoyi Zhang, Zongyu Guo, Yuan Zhang, Houqiang Li, and Yan Lu. CoD-Lite: Real-time diffusion-based generative image compression. *arXiv preprint arXiv:2604.12525*, 2026.

Tianyu Zhang, Xin Luo, Li Li, and Dong Liu. StableCodec: Taming one-step diffusion for extreme image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Naifu Xue, Zhaoyang Jia, Jiahao Li, Bin Li, Yuan Zhang, and Yan Lu. OneDC: One-step diffusion-based image compression with semantic distillation. In *Advances in Neural Information Processing Systems*, 2025b.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *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. In *International Conference on Learning Representations*, 2023.

Anaïs Mercat, Miska Viitanen, and Jarno Vanne. UVG dataset. *Zenodo*, 2020. doi:10.5281/zenodo.3885383.

Shen Wang, Amir Rehman, Zhou Wang, Kede Ma, and Guangtao Zhai. Mcl-JCV: a JND-based H.264/AVC video quality assessment dataset. *IEEE International Conference on Image Processing*, pages 41–45, 2016.

Yifan Bian, Chuanbo Tang, Li Li, and Dong Liu. Augmented deep contexts for spatially embedded video coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2094–2104, 2025.

Xingchen Li, Junzhe Zhang, Junqi Shi, Ming Lu, and Zhan Ma. YODA: Yet another one-step diffusion-based video compressor. *arXiv preprint arXiv:2601.01141*, 2026.

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/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

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*, 44(5):2567–2581, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Duolikun Danier, Fan Zhang, and David Bull. FloLPIPS: A bespoke video quality metric for frame interpolation. *arXiv preprint arXiv:2207.08119*, 2022.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *European conference on computer vision*, pages 179–195. Springer, 2018.

Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *ITU SG16 Doc. VCEG-M33*, 2001.

Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2535–2545, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

A Appendix

A.1 Mathematical Details

Derivation of $t^\star$:

We assume that the compression noise introduced by the compressor $\mathcal{C}(\cdot)$ based on the variational structure [Ballé et al., 2016] follows a Gaussian distribution with zero mean, i.e.:

$$\mathcal{C}(z) = z_c \sim \mathcal{N}\left(\boldsymbol{z}, \sigma^2\right) \tag{11}$$

For a given time step t, we can compute the flow state at this time step according to the definition of normalizing flows [Lipman et al., 2023]:

$$\mathbf{FL}(t) = (1-t)z + t\boldsymbol{\epsilon} \tag{12}$$

Here, $\boldsymbol{\epsilon}$ follows the standard normal distribution $\mathcal{N}(0, \mathbf{I})$, and thus $\mathbf{FL}(t)$ also follows a normal distribution: $\mathbf{FL}(t) \sim \mathcal{N}((1-t)z, t^2\mathbf{I})$.

The problem discussed in the main text is to solve for $t^\star$:

$$t^* = \arg\min_t \mathrm{dist}\left[(1-t)z_c, \mathbf{FL}(t)\right], t \in [0,1] \tag{13}$$

Here, $\mathbf{dist}(\cdot,\cdot)$ denotes the distance metric. We adopt the Wasserstein Distance, and the minimization problem is converted into finding the minimum value of the following expression:

$$f(t) = (1-t)^2(z-z)^2 + (|1-t|\sigma - |t|)^2 \tag{14}$$

$$= ((1-t)\sigma - t)^2 \tag{15}$$

It is easy to see that the minimum of $f(t)$ is $f(t^\star) = 0$, which occurs at:

$$t^\star = \frac{\sigma}{1+\sigma} \tag{16}$$

In our experiments, we approximate σ with the sample variance to estimate $t^\star$.

A.2 Additional Setting Details

A.2.1 Model Traning Details

Our model uses Wan2.1 1.3B [2] as the foundation flow model, with the context model and entropy model based on modules from DCVC and ELIC. Training is divided into two stages. The first stage does not consider inter-group information flow, i.e., training is conducted in the I-Group mode. In this stage, we use a GOP size of 5 and train on the Vimeo-90k dataset [Xue et al., 2019]. In the second stage, we expand the GOP size to 9 and set the number of frames within a meta-group to 17 (i.e., one I-Group and one P-Group), and fine-tune on the YouHQ dataset [Zhou et al., 2024]. The patch size is set to 256×256 throughout training. During training, we set the hyperparameters in the training objective as: $\gamma_1, \gamma_2, \gamma_3 = 2, 0.2, 0.1$. We adjust $\lambda_1, \lambda_2 = \{1,6\}, \{1,12\}, \{1,25\}, \{1,45\}$ to obtain different rate-distortion trade-offs.

Throughout all training stages, we employed AdamW [Loshchilov and Hutter, 2017] as the optimizer. The learning rates were strategically set at 1×10^{-4} for the initial phase and adjusted to 5×10^{-5} for the subsequent phase to facilitate finer parameter tuning. For `VoRTeC`$^+$, we perform fine-tuning using LoRA, where the rank of LoRA is set to 8. All experiments were conducted on an Nvidia A6000 GPU.

A.2.2 Test setting Detail

Original videos are stored in YUV420 format and converted to RGB according to the BT.709 standard. For codecs that require the input resolution to be a multiple of 64, zero-padding is applied before encoding; after decoding, the decoded frames are cropped to restore the original spatial dimensions. For 720p testing, we center-crop the original 1080p data to 720p.

[2] https://huggingface.co/Wan-AI/Wan2.1-T2V-1.3B

Traditional Video Codecs :

We evaluate VTM-17.0 [Bross et al., 2021] in the RGB color space, adopting the settings from the DCVC series [Li et al., 2021], converting the RGB input to 10-bit YUV444 format for internal codec processing. The configuration files used for each codec are as follows.

- VTM: `encoder_lowdelay_vtm.cfg`

The used parameters for each video as:

- `-c {config file name}`
- `-InputFile={input video name}`
- `-InputBitDepth=10`
- `-OutputBitDepth=10`
- `-OutputBitDepthC=10`
- `-InputChromaFormat=444`
- `-FrameRate={frame rate}`
- `-DecodingRefreshType=2`
- `-FramesToBeEncoded={frame number}`
- `-SourceWidth={width}`
- `-SourceHeight={height}`
- `-IntraPeriod={intra period}`
- `-QP={qp}`
- `-Level=6.2`
- `-BitstreamFile={bitstream file name}`

Neural Codecs :

DCVC-FM, DCVC-RT: We use the officially released source code and pretrained weights [3].

GLC-Video: We implement it using the open-source code and pretrained weights provided by GLC-Video [4].

SEVC: We implement it using the open-source code and pretrained weights provided by SEVC [5].

GNVC-VD, YODA, DiffVC, FreeGVC: Since these works are not open-sourced, we use the original data reported in their respective papers while ensuring consistent experimental settings.

S2VC: We use the bitrate data directly provided by the original authors of S2VC.

For LPIPS, we utilized the *lpips* library, while DISTS was implemented using *DISTS_pytorch*. FID metric were calculated using functions provided by *torchmetrics.image*, with a feature size of 2048. Following the approach in [Mentzer et al., 2020], during FID testing, images were partitioned. Specifically, from each $H \times W$ image, we initially extracted $\lfloor H/f \rfloor \cdot \lfloor W/f \rfloor$ non-overlapping $f \times f$ crops. Subsequently, the extraction origin was shifted by $f/2$ in both dimensions to obtain another $(\lfloor H/f \rfloor - 1) \cdot (\lfloor W/f \rfloor - 1)$ patches. A fixed value of $f = 256$ was used for all evaluations.

A.3 Additional Experiment Results

In this section, we present additional quantitative experimental results, including datasets and evaluation metrics not included in the main text. Furthermore, we conduct comprehensive visual comparisons between `VoRTeC` and the baseline methods.

A.3.1 Quantitative Results

We first present the quantitative comparison results on the HEVC-Class C and MCL-JCV datasets, as illustrated in Figure 8 and Figure 9. Consistent with the observations and discussions in the main text, `VoRTeC` achieves superior

[3] https://github.com/microsoft/DCVC
[4] https://github.com/jzyustc/GLC
[5] https://github.com/EsakaK/SEVC

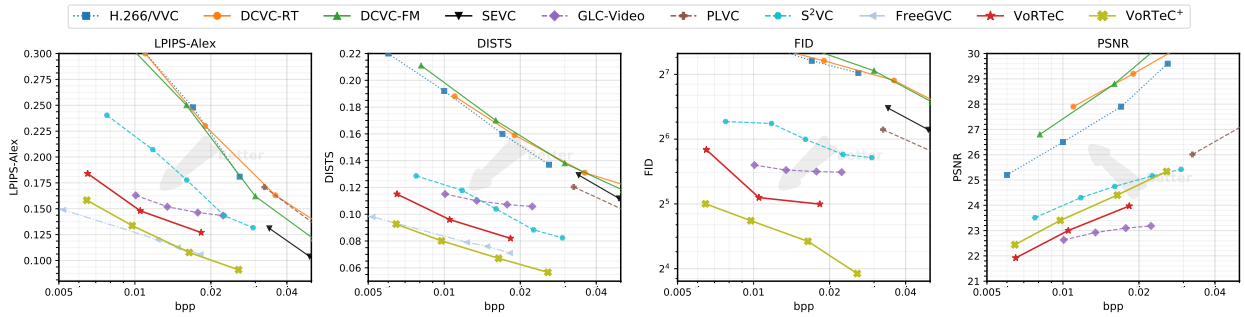


Figure 8: Rate–distortion/perception curves on HEVC Class C datasets.

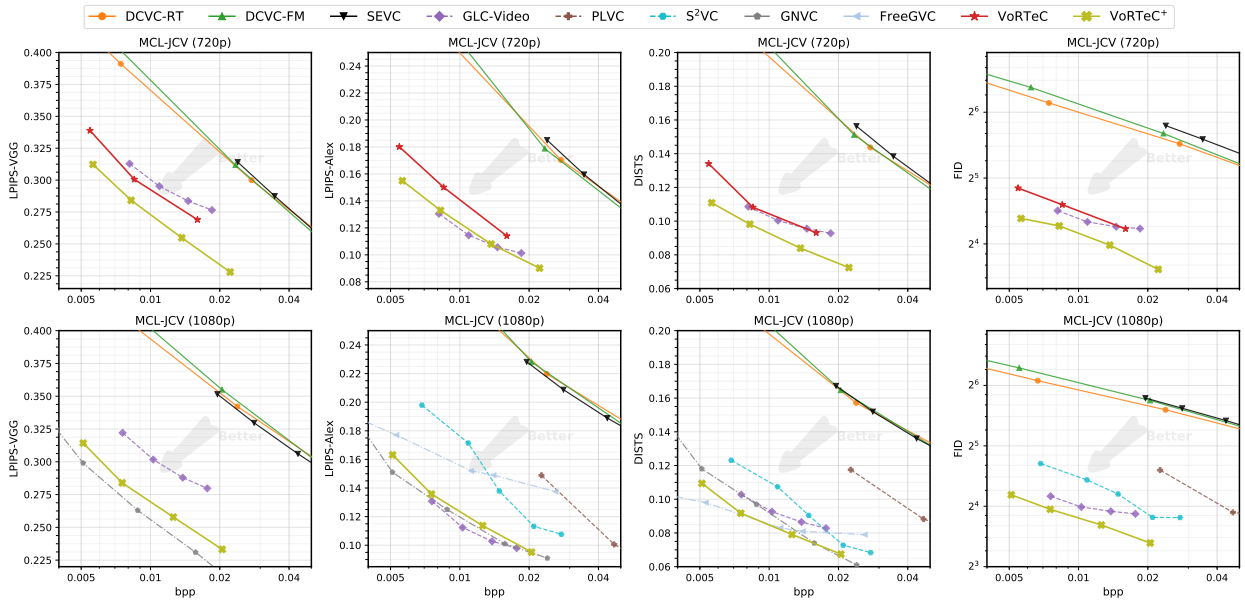


Figure 9: Rate–perception curves on MCL-JCV datasets.

performance on low-resolution (480p) data, consistently outperforming generative methods across all metrics. On the MCL-JCV dataset, our proposed solution also exhibits strong competitiveness. It consistently surpasses GLC-Video on all metrics and delivers comparable performance to GNVC-VD.

Figure 10 reports comparative experiments for all remaining distortion metrics over the HEVC-Class B, UVG, and MCL-JCV datasets. This figure validates the excellent compression capability of VoRTeC, which yields considerably higher PSNR than GLC-Video.

In addition, we report the BD-Rate measured on all datasets (with `VoRTeC` as the anchor), as shown in Table 3. For some methods, the RD curves do not overlap with that of `VoRTeC`, making it infeasible to compute the BD-Rate values.

A.3.2 Ablation Result

Prior ablation: Figure 11 shows the ablation on bitrate allocation across modules. (a) shows the bitrate allocation map learned by the latent codec when the flow prior is removed, while (b) shows the bitrate allocation map after incorporating the flow prior. As can be seen, since `VoRTeC` removes the prior redundancy of the codec through end-to-end training, the bitrate is more accurately and reasonably allocated to specific objects rather than to flat background regions.

Prior cahce ablation: Figure 13 demonstrates the contribution of the prior cache to inter-frame consistency. Without the prior cache, group transitions cause noticeable inconsistencies, manifesting as changes in texture, color, and shape (e.g., the color and size of the eyes), whereas the introduction of the prior cache effectively resolves this issue.

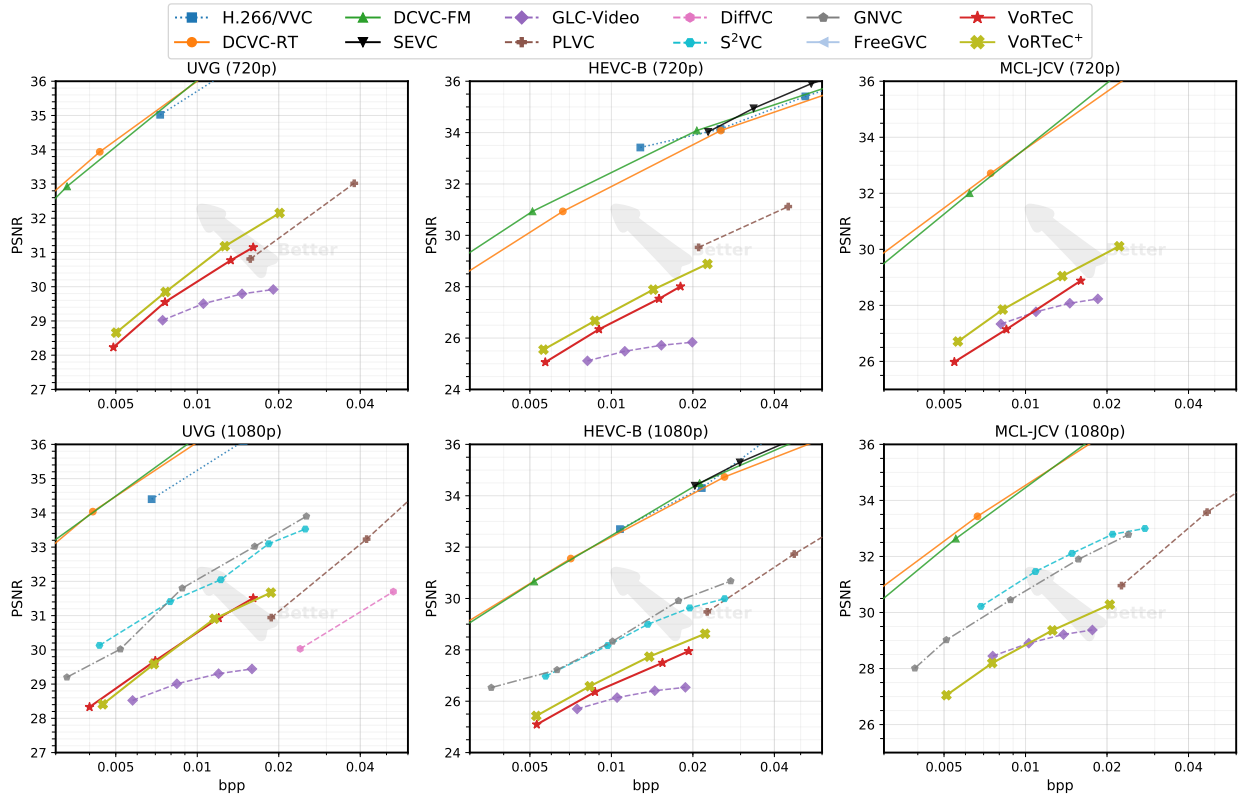
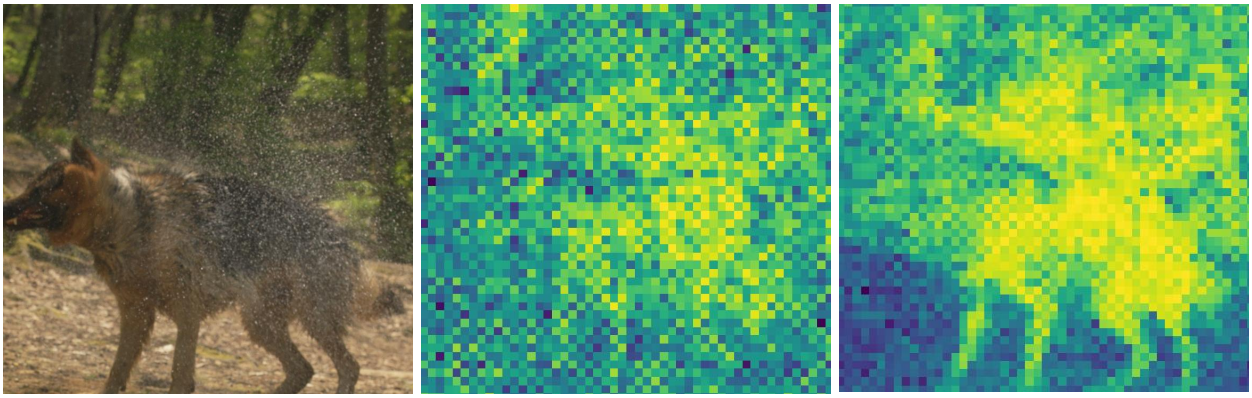


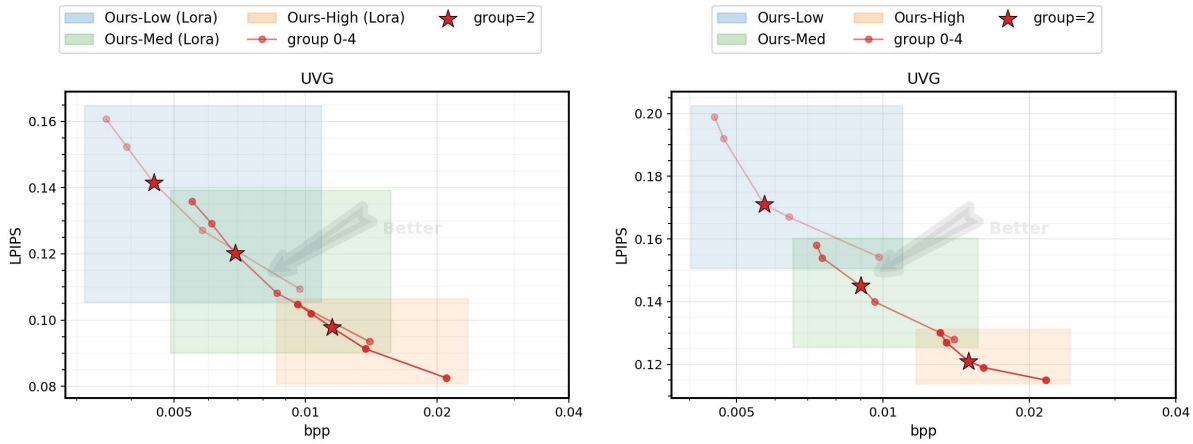
Figure 10: Rate–distortion curves on HEVC Class B, HEVC Class C, UVG and MCL-JCV datasets.



(a) Original (b) W/o Flow Prior (c) W/ Flow Prior

Figure 11: Investigate the impact of the flow prior on learning the codec bit-rate distribution. Incorporating the flow prior enables the codec to learn more compact and rational bit allocation during end-to-end training.

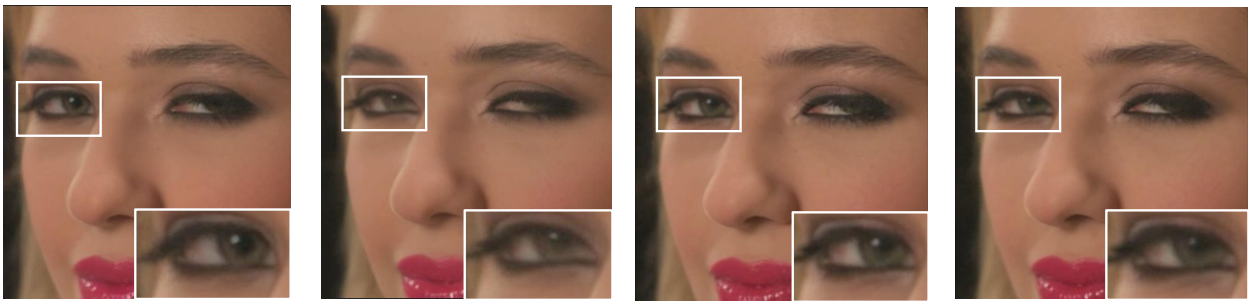
Meta Group ablation: We conduct an ablation on the meta-group size. Since our designed CGG (Contact Group to Group) is scalable, we can increase the ratio of P-Groups to I-Groups to achieve bitrate adjustment within a certain range. In the main text, we set this ratio to 2, meaning that a meta-group contains 25 frames (9 + 8 + 8), consisting of one I-Group and two P-Groups. Figure 12 demonstrates the scalability of our models at different bitrates. Reducing the number of P-Groups leads to an increase in bitrate, which in turn improves reconstruction quality. This indicates that our approach can flexibly cover a wider range of bitrates.



(a) Ablation study on our LoRA fine-tuning scheme

(b) Ablation study of our proposed scheme

Figure 12: We achieve a certain range of bit-rate scaling by adjusting the number of Pgroups within one Meta-group. Each color indicates the scalable bit-rate range supported by our single model, and the red star marks the number of Pgroups adopted in our main experiments. This further demonstrates the scalability and flexibility of `VoRTeC`.



(a) w/o prior cache G=0 (t=9)

(b) w/o prior cache G=1 (t=0)

(c) w/ prior cache G=0 (t=9)

(d) w/ prior cache G=1 (t=0)

Figure 13: The impact of prior caching on inter-group consistency is investigated. When prior caching is disabled, learning inter-group contextual information becomes difficult, resulting in abrupt changes to details including colors and contours.

Table 3: BD-rate (%) comparison on five datasets. All values are computed relative to VoRTeC (Ours). Positive values indicate that the corresponding method requires a higher bitrate than VoRTeC to achieve the same quality, *i.e.*, VoRTeC is superior. "–" denotes unavailable results.

Method	$LPIPS_{Alex}$	$LPIPS_{VGG}$	DISTS	FID	PSNR
(a) UVG 720p					
GLC	-5.81	58.69	13.18	-1.61	43.70
DCVC-RT	–	340.98	–	–	**-88.45**
DCVC-FM	–	344.47	–	–	-85.06
SEVC	–	335.20	–	–	–
VTM	1302.42	925.28	899.59	–	–
PLVC	156.09	158.04	138.70	283.08	13.80
VoRTeC	0.00	0.00	0.00	0.00	0.00
VoRTeC$^+$	**-12.32**	**-30.44**	**-37.21**	**-45.32**	-13.49
(b) HEVC-B 720p					
GLC	15.15	61.58	45.95	24.46	75.61
DCVC-RT	1332.36	408.84	1217.26	1573.37	-86.42
DCVC-FM	1187.45	426.88	1184.74	–	**-89.22**
SEVC	1171.48	383.14	1401.63	–	–
VTM	1062.20	381.97	763.14	2059.48	–
PLVC	189.02	184.93	180.85	130.14	–
VoRTeC	0.00	0.00	0.00	0.00	0.00
VoRTeC$^+$	**-26.94**	**-33.94**	**-44.38**	**-51.20**	-16.57
(c) HEVC Class C					
GLC	39.81	95.75	90.45	77.14	36.07
SEVC	103.04	201.26	570.04	1191.85	–
S2VC	113.87	–	80.88	244.79	**-45.63**
PLVC	301.00	–	476.05	925.16	–
DCVC-FM	239.47	–	–	–	–
DCVC-RT	319.31	–	–	–	–
VTM	293.43	–	–	–	–
FreeGVC	**-41.37**	–	-44.21	–	–
VoRTeC	0.00	0.00	0.00	0.00	0.00
VoRTeC$^+$	-30.89	**-39.20**	**-47.56**	**-64.07**	-23.88
(d) HEVC-B 1080p					
GLC	-4.22	60.55	20.14	5.01	36.11
DCVC-RT	1195.73	452.70	1121.76	1651.50	**-89.72**
DCVC-FM	1028.40	454.92	1032.56	–	-88.07
SEVC	1093.06	455.19	1072.35	–	–
VTM	976.40	364.72	605.10	572.67	–
S2VC	106.50	–	40.10	94.19	-53.02
PLVC	119.76	–	202.20	108.71	–
GNVC	-21.12	**-34.58**	-30.47	–	-44.96
FreeGVC	116.59	–	-40.92	–	–
VoRTeC	0.00	0.00	0.00	0.00	0.00
VoRTeC$^+$	**-25.73**	-34.40	**-46.71**	**-46.52**	-15.49
(e) UVG 1080p					
GLC	6.02	78.64	22.55	34.99	63.53
DCVC-RT	–	393.69	–	–	**-88.64**
DCVC-FM	–	388.62	–	–	-87.01
SEVC	–	388.43	–	–	–
VTM	1430.52	350.12	858.28	–	–
DiffVC	273.79	–	178.37	–	203.22
S2VC	138.78	–	99.59	134.30	-48.81
PLVC	79.10	–	236.45	309.67	49.87
GNVC	5.83	**-23.88**	1.63	–	-40.55
FreeGVC	58.56	–	4.88	–	–
VoRTeC	0.00	0.00	0.00	0.00	0.00
VoRTeC$^+$	**-5.36**	-21.22	**-19.33**	**-17.88**	-1.80