Active Sampling for Ultra-Low-Bit-Rate Video Compression via Conditional Controlled Diffusion

Anonymous authors
Paper under double-blind review

Abstract

Diffusion models provide a powerful generative prior for perceptual reconstruction at ultra-low bitrates, but effective video compression requires controlling the generative process using highly compact conditioning signals. In this work, we present **ActDiff-VC**, a diffusion-based video compression framework for the ultra-low-bitrate regime. Our method partitions videos into variable-length segments, transmits keyframes only when needed, and summarizes temporal dynamics using a compact set of tracked point trajectories. Conditioned on these sparse signals, a conditional diffusion decoder synthesizes the remaining frames, enabling perceptually realistic reconstruction under severe rate constraints. To support this design, we introduce two mechanisms: content-adaptive keyframe selection and budget-aware sparse trajectory selection, which together enable compact yet effective conditioning for generative reconstruction. Experiments on the UVG and MCL-JCV benchmarks show that ActDiff-VC achieves up to 64.6% bitrate reduction at matched NIQE, improves KID by up to 64.6% and FID by up to 37.7% at comparable bitrates against strong learned codecs, and delivers favorable perceptual rate–distortion trade-offs relative to learned and diffusion-based baselines in the ultra-low-bitrate regime.

1 Introduction

As video increasingly underpins everyday communication and entertainment, it has become a dominant contributor to network traffic and storage demand. This scale makes improvements in compression efficiency essential, especially as expectations for visual quality continue to rise while bandwidth budgets continue to shrink. The central challenge is to deliver higher perceived quality under tighter rate constraints. Learned video compression has advanced primarily through distortion-minimizing training, but these principles strain in the ultra-low-bitrate regime. Here, we use ultra-low bitrate to refer to operating points roughly at or below 0.05 bits per pixel, where the transmitted information per frame is severely constrained. At such rates, the encoder can no longer afford to transmit rich motion side information or detailed residual corrections, and reconstructions often become over-smoothed. In this regime, perceived quality is governed more by realism than by pixel-level fidelity, motivating decoders that rely on powerful generative priors to infer missing content from minimal side information.

Diffusion models provide a powerful generative prior for reconstruction at ultra-low bitrates, enabling perceptually realistic synthesis from highly compressed conditioning signals. However, integrating diffusion into video compression raises practical challenges in controlling the generative process under tight rate budgets. Rich conditioning can improve reconstruction fidelity but increases side-information cost, while overly sparse conditioning can lead to perceptually implausible or unstable reconstructions. Existing diffusion-based codecs address this trade-off in different ways. I^2VC Liu et al. (2024) leverages latent diffusion priors but performs iterative denoising independently for each frame, limiting efficiency. EVC-PDM Li et al. (2024a) evaluates candidate reconstructions by running diffusion at the encoder to determine when to transmit additional keyframes, incurring significant encoder overhead. These methods highlight the central challenge of designing compact yet informative conditioning mechanisms that enable high perceptual reconstruction quality in ultra-low-bitrate diffusion-based video compression.

To address this conditioning efficiency challenge, we introduce a generative compression strategy that exploits temporal redundancy through structured sparse conditioning. Instead of reconstructing frames via per-frame diffusion reconstruction, we partition the video into variable-length segments and transmit keyframes when the current keyframe is no longer sufficiently informative for reconstructing subsequent frames. Within each segment, temporal dynamics are summarized by a compact set of tracked point trajectories that serve as motion conditioning signals for generative reconstruction. A conditional diffusion decoder then synthesizes the remaining frames from these sparse signals, enabling perceptually realistic reconstruction at ultra-low bitrates. Our approach is motivated by the observation that motion in natural videos exhibits strong spatial correlation, allowing sparse trajectory conditioning to approximate dense motion fields with minimal perceptual degradation. This enables a compression design in which a small number of transmitted keyframes and compact motion cues are sufficient to guide generative reconstruction of intermediate content. Building on this principle, we leverage pre-trained diffusion models to realize sparse motion-guided video reconstruction in the ultra-low bitrate regime, and summarize our contributions as follows:

- We introduce a generative video compression framework in which a conditional diffusion decoder reconstructs video from *sparse trajectory conditioning*, enabling perceptually realistic reconstruction at ultra-low bitrates.
- We propose two encoder-side mechanisms for compact and informative conditioning: (i) *content-adaptive keyframe selection* that forms variable-length segments aligned with scene dynamics, and (ii) a *budget-aware sparse trajectory selection* strategy that extracts representative motion cues from dense tracking to minimize side-information rate.
- We demonstrate strong perceptual compression performance in the ultra-low-bitrate regime on the UVG and MCL-JCV benchmarks, achieving up to 64.6% bitrate reduction at matched NIQE, improving KID by up to 64.6% and FID by up to 37.7% at comparable bitrates against strong learned codecs. We further show favorable perceptual rate–distortion trade-offs relative to learned and diffusion-based baselines, and extensive ablations validate the contribution of each component.

2 Related work

Learning-based video codecs have achieved substantial gains over traditional standards by replacing hand-crafted modules with neural networks (Wiegand et al., 2003; Sullivan et al., 2012). Most early approaches follow a predictive residual-coding paradigm, where motion-compensated predictions are formed from previously decoded frames and the encoder transmits compact residual representations (Lu et al., 2019). Subsequent methods extend this framework through conditional coding (Li et al., 2021; 2022; 2023), which models an explicit distribution of the current frame conditioned on decoded history and entropy-codes it under the learned distribution. A complementary direction is interpolation-based compression (Wu et al., 2018; Alexandre et al., 2023), which synthesizes intermediate frames from sparsely transmitted reference frames. While these learned codecs achieve strong distortion performance at moderate bitrates, they are typically trained with per-frame distortion objectives that lead to over-smoothed and perceptually blurry reconstructions in the ultra-low-bitrate regime (Blau & Michaeli, 2019). This limitation motivates compression strategies that prioritize perceptual realism over strict pixel fidelity under extreme rate constraints.

Perceptual video compression improves visual quality by augmenting learned codecs with perceptual losses and adversarial training, thereby biasing reconstructions toward human visual preferences rather than exact pixel fidelity (Xu et al., 2024). GAN-based methods further enhance high-frequency details and textures through adversarial objectives (Veerabadran et al., 2020; Du et al., 2022; 2024). However, existing perceptual and GAN-based codecs do not fully address the ultra-low-bitrate regime (e.g., ≤ 0.05 bpp). In this range, the bit budget severely limits the information that can be transmitted per frame, and distortion-optimized learned codecs often produce noticeably blurry reconstructions. More critically, many perceptual and GAN-based codecs determine their operating points through training-time choices, such as rate–distortion loss weights or separately trained rate models, making them difficult to adapt reliably to previously unseen ultra-low-bitrate settings without retraining. This gap motivates leveraging stronger generative priors that can reconstruct realistic video from highly compact conditioning signals.

Diffusion models (Ho et al., 2020) have emerged as powerful generative priors capable of synthesizing photorealistic content from compact conditioning signals, including text (Nichol et al., 2021; Ho et al., 2022), sketches (Zhang et al., 2023), and point trajectories (Gu et al., 2025). This capability makes them well suited for ultra-low-rate video compression, where the bit budget for transmitted side information is severely constrained. In the image domain, diffusion-based codecs have demonstrated substantial perceptual gains at very low bitrates (Pan et al., 2022; Yang & Mandt, 2023; Lei et al., 2023; Careil et al., 2023; Relic et al., 2024), indicating that diffusion priors can improve reconstruction quality under extreme rate constraints. Extending these benefits to video, however, remains challenging, as reconstruction must both exploit temporal redundancy and operate under highly compressed conditioning.

Recent methods have begun incorporating diffusion priors into learned video compression. I^2VC (Liu et al., 2024) encodes frames into latent representations, allocates bits using a learned spatio–temporal importance mask, and reconstructs them with a pre-trained image latent diffusion model conditioned on features from neighboring decoded frames. However, without an explicit motion signal, maintaining consistent reconstruction under large motion or occlusion may be challenging. EVC-PDM (Li et al., 2024a) instead transmits a subset of intra frames and synthesizes skipped frames with a video diffusion model. It selects keyframes by performing diffusion-based forecasting at the encoder and inserting a keyframe when predicted perceptual quality drops below a threshold. However, its keyframe-selection procedure relies on repeatedly running diffusion-based forecasts at the encoder to decide when additional intra frames are required, incurring substantial encoder-side overhead and limiting practicality. These limitations motivate approaches that avoid repeated diffusion-based forecasting at the encoder and instead use compact sparse trajectories as direct conditioning for diffusion-based reconstruction.

Adaptive keyframe placement addresses temporal variation by selecting content-dependent segments instead of relying on fixed group-of-pictures (GOP) structures. Most learned video codecs use a fixed GOP, which can be suboptimal when motion intensity, scene dynamics, or shot boundaries vary over time. Adaptive strategies form variable-length segments that better allocate bits across the video (Ge et al., 2024; Yang et al., 2024). The most relevant work is M3-CVC (Wan et al., 2025): it segments video into variable-length clips based on semantic changes and optical flow, and reconstructs each clip from a decoded keyframe together with a text description using a diffusion-based generative model. While promising, text-based conditioning can be too coarse or ambiguous to capture fine-grained motion and complex dynamics, motivating the use of compact sparse trajectories as a more explicit conditioning signal for diffusion-based reconstruction.

Domain-specific ultra-low-bitrate video communication has also been explored using compact semantic control signals, such as facial landmarks or keypoints for conferencing (Oquab et al., 2021; Konuko et al., 2021) and parametric human-body representations (Chen et al., 2025). These approaches exploit strong structural priors about the content (e.g., faces or articulated bodies) to achieve extreme compression. However, because they rely on domain-specific assumptions, they are not directly comparable to methods targeting unconstrained natural videos, which is the setting we consider.

3 Proposed Approach: Concept Overview

In this section, we provide a conceptual overview of our proposed generative video compression algorithm. We couple *active sampling* of keyframes and intermediate context at the encoder with a *conditional diffusion* decoder. The sampled keyframes and selected context create a content-adaptive temporal segmentation of the video where the decoding is done in blocks of variable length. The decoder treats the selected context as side information and reconstructs the intermediate frames between each pair of consecutive keyframes by running a reverse diffusion process conditioned on this received side information (in the form of sparse trajectory conditioning). Figure 1 illustrates our proposed pipeline.

3.1 Detailed Overview

We decompose each input video into a sequence of *content-adaptive* segments, also known as groups of pictures (GOPs), indexed by $k = 1, \ldots, K$. GOP segment k contains T_k frames $\{x_t^{(k)}\}_{t=1}^{T_k}$ with $x_t^{(k)} \in \mathbb{R}^{3\times H\times W}$ and a one-frame overlap with the preceding and subsequent segments, i.e., $x_{T_k}^{(k)} = x_1^{(k+1)}$. At a high level, the

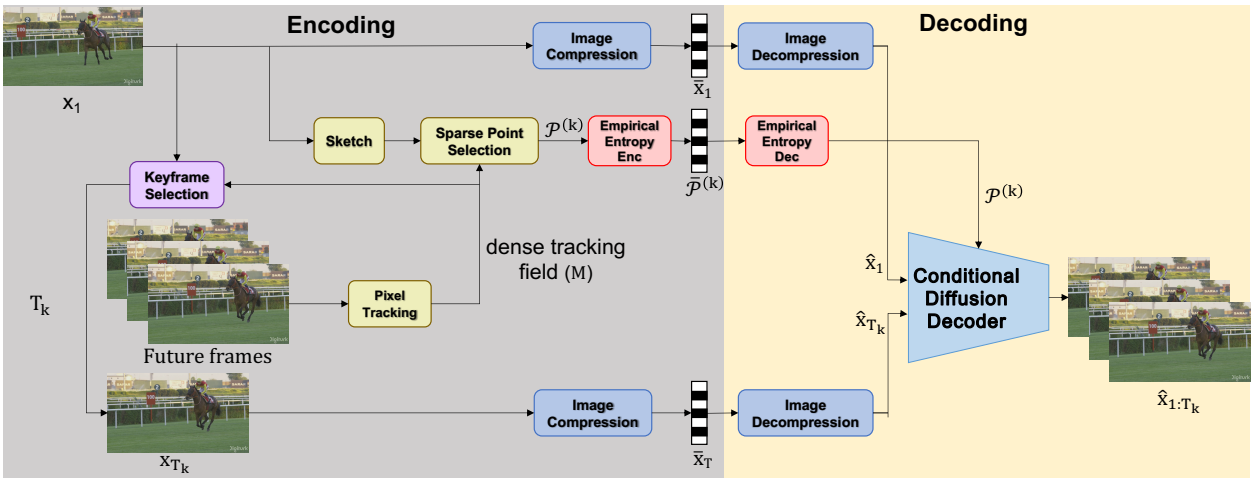

Figure 1: **Framework of ActDiff-VC.** Given the first frame, a dense point tracker estimates the dense tracking field $\mathbf{M}$ across subsequent frames. The sparse point selector, guided by a sketch of the first frame, subsamples the dense tracking field to form the conditioning sparse trajectory set $\mathcal{P}^{(k)}$. On the decoder side, the diffusion model is conditioned on $\mathcal{P}^{(k)}$ together with the first and last frames to reconstruct the video.

encoder observes the video stream and chooses when to sample densely through keyframes and when to sample sparsely through compact motion conditioning signals. The decoder, on the other hand, relies on a conditional diffusion model to generate the missing information across each segment.

To guide both sampling decisions and the diffusion decoder, we use motion information from video tracking. Let $\Omega \subset \mathbb{Z}^2$ denote the set of pixel coordinates in the first frame $x_1^{(k)}$. The dense point tracker estimates, for each pixel $p \in \Omega$, a sequence of 2D displacements from the initial frame at each subsequent frame t:

$$\mathbf{M} : \Omega \to \mathbb{R}^{2T_k}, \quad p \mapsto \mathbf{M}(p) \equiv \big(\mathbf{u}_1(p), \ldots, \mathbf{u}_{T_k}(p)\big), \tag{1}$$

where $\mathbf{u}_t(p) \in \mathbb{R}^2$ denotes the displacement vector of pixel p from its position in the initial frame $x_1^{(k)}$ to its position at frame t. This dense tracking field $\mathbf{M}$ captures the motion and temporal variation throughout the k-th segment, allowing the model to represent both local and global scene dynamics. While pixel tracking offers several advantages over traditional motion representations, its most important advantage for us is the high spatial correlation. More specifically, in natural videos, the trajectory of the pixel p, $p \mapsto \mathbf{M}(p)$, is highly correlated with the trajectory of its neighboring pixel q, $q \mapsto \mathbf{M}(q)$. This inherently high spatial correlation allows for further sub-sampling of the trajectory to a sparse pixel set $S \subset \Omega$ without negatively impacting motion fidelity while guaranteeing an ultra-low bit rate. We therefore introduce the *sparse trajectory set* for the k-th segment as the subsampling of the dense tracking field $\mathbf{M}$ onto S,

$$\mathcal{P}^{(k)} \;:=\; \big\{\, \big(q, \{\mathbf{u}_t(q)\}_{t=1}^{T_k}\big) \;:\; q \in S \,\big\}. \tag{2}$$

Our conditional diffusion model decoder operates on each segment independently, relying on the frame overlaps to effectively regulate the temporal variations across GOP segments. Given the decoded boundary keyframes $\hat{x}_1^{(k)}$, $\hat{x}_{T_k}^{(k)}$ and sparse trajectory set $\mathcal{P}^{(k)}$, , our decoder synthesizes a length-T_k sequence of frames whose appearance matches $\hat{x}_1^{(k)}$, $\hat{x}_{T_k}^{(k)}$ and whose motion follows $\mathcal{P}^{(k)}$. For notational simplicity, we omit segment superscripts on segment-dependent quantities whenever the segment index is clear from context. The final component handles compression. Keyframes are compressed with a lossy image compressor, whereas the sparse trajectory set is compressed losslessly with an entropy coder. The transmitted bitstream therefore comprises (i) compressed keyframes, (ii) losslessly compressed sparse trajectory set, and (iii) the segment size. At the receiver, the decoder first decompresses the keyframes and the sparse trajectory set; it then uses the keyframes together with $\mathcal{P}^{(k)}$ to guide diffusion-based synthesis of each segment. The following subsections describe each component conceptually, while Section 4 describes our concrete implementation.

3.2 Video Tracking as Sparse Trajectory Conditioning

To guide the diffusion decoder's motion synthesis, we represent temporal dynamics using dense pixel tracking. Rather than transmitting conventional motion vectors or optical flow fields, we leverage recent advances in dense point tracking (e.g., (Harley et al., 2025)) to estimate a dense tracking field $\mathbf{M}$ over each segment. These trajectories capture how each pixel in the reference keyframe $x_1^{(k)}$ moves through subsequent frames, providing a rich motion signal that the diffusion model can condition on during generation. Pixel tracking offers several advantages over traditional motion representations. First, tracking naturally handles long-range motion and maintains temporal consistency across multiple frames, whereas frame-to-frame motion vectors can accumulate drift (Neoral et al., 2024; Harley et al., 2025). Second, the dense tracking field provides explicit correspondence information that helps the diffusion model understand which regions of the reference keyframe $x_1^{(k)}$ should appear at which locations in future frames, reducing ambiguity in the generation process. Furthermore, tracking is inherently sparse-compatible. That is, given a sparse set S of selected points in the initial frame, we estimate dense displacements via a radial basis function (RBF) kernel:

$$\widehat{\mathbf{M}}(p \mid S) = \sum_{q \in S} \alpha(p,q)\, \mathbf{M}(q), \qquad \alpha(p,q) = \frac{\exp\Big(-\frac{\|p-q\|^2}{2\sigma^2}\Big)}{\sum_{q' \in S} \exp\Big(-\frac{\|p-q'\|^2}{2\sigma^2}\Big)}. \tag{3}$$

where $\|p-q\|$ denotes the Euclidean distance between pixels p and q, and σ is the kernel bandwidth. We assume local spatial correlation in the tracking field: pixels that are close in the initial frame $x_1^{(k)}$ are more likely to have similar motion trajectories, especially when they lie in the same motion region. We can find $S \subset \Omega$ with $|S| \leq B$ such that the weighted reconstruction error:

$$\mathcal{R}(S) \;\equiv\; \sum_{p \in \Omega} w^{(k)}(p)\, \big\|\mathbf{M}(p) - \widehat{\mathbf{M}}(p \mid S)\big\|_2^2, \tag{4}$$

can be small, where $w^{(k)} : \Omega \to \mathbb{R}_{\geq 0}$ is an importance weight derived from the first frame (e.g., edge magnitude or gradient saliency).

3.3 Conditional Diffusion Decoder

Our compression framework relies on a conditional video diffusion model that serves as the decoder. Unlike traditional codecs that reconstruct frames through explicit transform coding of residuals, our approach leverages the generative capabilities of diffusion models to synthesize video content from minimal conditioning signals. Ideally, such a conditional diffusion model is trained from scratch; however, it can also be built by adapting existing diffusion-based video-generation models. Section 4.1 provides a detailed description of this model and how we adapt it for video compression at ultra-low bit rate.

Our proposed diffusion model takes as input: (i) conditioning image latents from the first and last frames of each GOP segment, providing appearance anchors, and (ii) the sparse trajectory set $\mathcal{P}^{(k)}$ that encodes motion dynamics throughout the temporal extent of the segment. The diffusion process iteratively refines an initial noisy latent representation toward a high-quality video sequence that matches both the appearance constraints (from keyframes) and the motion constraints (from tracking). By conditioning the diffusion process on both appearance and motion, the model can generate high-fidelity video sequences that respect the scene's motion structure while maintaining visual fidelity to the keyframe appearances. Furthermore, frame overlap ensures smooth transitions across consecutive GOP segments.

3.4 Content-Adaptive Keyframe Selection

A fundamental challenge in video compression is determining optimal keyframe placement. Traditional codecs often use fixed GOP structures with predetermined keyframe intervals, which fail to account for varying scene complexity, motion patterns, and visual content across different video segments.

Our approach introduces content-adaptive keyframe selection that determines GOP segment boundaries based on how long appearance and motion information from a keyframe remain informative for recon-

struction. In particular, given the dense tracking field $\mathbf{M}$ from the first frame of the segment $x_1^{(k)}$, we forward-splat the first frame $x_1^{(k)}$ along the displacements $\{\mathbf{u}_t(p)\}_{p\in\Omega}$ to obtain warped images $\tilde{x}_t^{(k)}$. The overlap between the warped and original frames, together with their perceptual similarity, is used to compute a keyframe-selection score θ_t, which serves as a surrogate for how informative the current keyframe remains for reconstructing future frames. The precise definition of θ_t is given in Section 4.

4 Proposed Approach: Implementation

This section describes the concrete implementation of the conceptual components introduced in Section 3. We begin by detailing the conditional video diffusion model and how it is adapted for compression. We then present our implementations of the content-adaptive keyframe selection, the corresponding budget-aware sparse trajectory selection, and encoding of all transmitted components.

4.1 Conditional Diffusion Decoder Implementation

Our conditional diffusion decoder is built upon DaS (Gu et al., 2025), a transformer-based video diffusion model that operates in a latent space. Given a conditioning image x_1 and a sparse trajectory set $\mathcal{P}$ of temporal length T, DaS synthesizes a length-T video whose appearance matches x_1 and whose motion follows $\mathcal{P}$. The model uses temporal and spatial downsampling factors $s_T = 4$ and $s_S = 8$. For notational clarity, we define $T_{\text{lat}} := \left\lceil \frac{T}{s_T} \right\rceil$, $h := \left\lceil \frac{H}{s_S} \right\rceil$, $w := \left\lceil \frac{W}{s_S} \right\rceil$. A motion VAE encoder $\mathcal{E}_{\text{trk}}$ maps the sparse tracking set $\mathcal{P}$ to a latent tensor $z_{\text{trk}} = \mathcal{E}_{\text{trk}}(\mathcal{P}) \in \mathbb{R}^{T_{\text{lat}}\times h\times w\times 16}$. The construction of this motion VAE encoder is described in Appendix A5. Separately, the VAE encoder $\mathcal{E}$ maps the conditioning image x_1 to a single-frame latent $z_1 = \mathcal{E}(x_1) \in \mathbb{R}^{1\times h\times w\times 16}$. The image latent is temporally expanded by placing it in the first latent time slice and zero-padding the remaining slices, yielding $z_{1,\text{pad}} \in \mathbb{R}^{T_{\text{lat}}\times h\times w\times 16}$. The denoising diffusion transformer (DiT) takes the concatenation of $z_{1,\text{pad}}$ with Gaussian noise and iteratively denoises it over multiple reverse diffusion steps. In parallel, a conditioning DiT processes the tracking latent z_{trk} and extracts motion features at each transformer block. These motion features are linearly projected and injected into the corresponding layers of the denoising DiT via additive conditioning, allowing the diffusion process to be guided by the sparse motion signal. Finally, the VAE decoder $\mathcal{D}$ maps the denoised latent back to pixel space to produce the reconstructed frames.

Bidirectional Conditioning. To mitigate temporal discontinuities at segment boundaries, we use the pretrained DaS model with bidirectional boundary conditioning at inference time, without additional training or fine-tuning. Let x_1 and x_{T_k} denote the segment's boundary frames, with corresponding latents z_1 and z_{T_k}. We form a *dual-conditioned latent stack*:

$$z_{\text{dual}} = \big[z_1, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{T_{\text{lat}}-2 \text{ interior zero slices}}, z_{T_k}\big], \tag{5}$$

where each zero slice $\mathbf{0} \in \mathbb{R}^{h\times w\times 16}$ has the same spatial and channel dimensions as a single latent frame. This dual-conditioned stack replaces the single-keyframe stack $z_{1,\text{pad}}$ used in the original DaS formulation. The denoising DiT then operates on the concatenation of z_{dual} with Gaussian noise, encouraging the generated sequence to remain consistent with both boundary keyframes. As adjacent segments share a one-frame overlap, this bidirectional conditioning introduces minimal rate overhead while improving temporal smoothness.

Adapting a fixed-length generator to variable-length segments. Although DaS produces a fixed-length sequence, our compression system operates on variable-length segments of length T_k. To reconcile these, we temporally interpolate the sparse trajectory set for segment k onto the model's fixed temporal grid. After generation, we subsample the generated sequence to T_k frames.

4.2 Pixel Tracking Implementation

Among candidate motion cues, including frame-to-frame optical flow, scene flow, and long-range point tracking, we adopt long-range point tracking to mitigate error accumulation from chaining short-horizon flows and to obtain temporally coherent trajectories over each segment. Specifically, we employ AllTracker (Harley

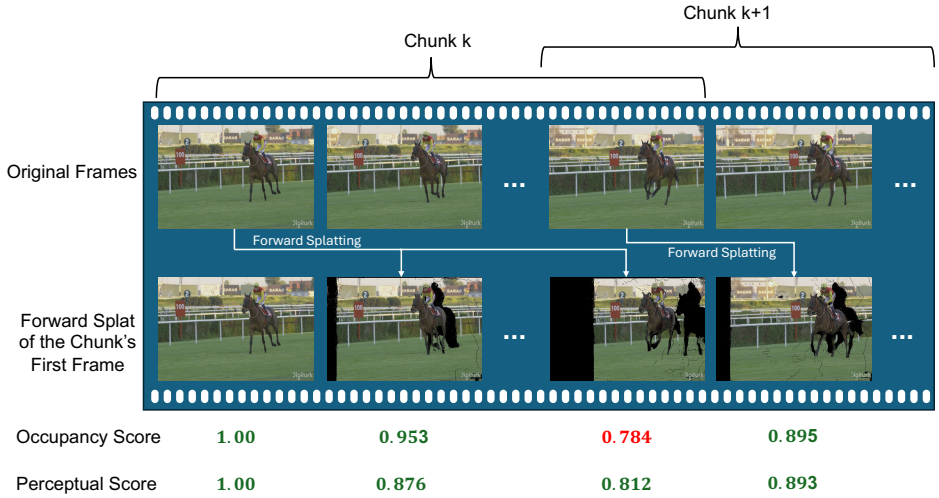

Figure 2: **Content-Adaptive Keyframe Selection.** The first frame of each segment is forward-splatted through the next frames using a dense tracker, yielding target-space occupancy $\mathrm{occ}(t)$ and perceptual similarity $\mathrm{sim}_{\mathrm{perc}}(t)$. The next keyframe is selected at the earliest t such that $\mathrm{occ}(t) < \theta_{\mathrm{occ}}$ or $\mathrm{sim}_{\mathrm{perc}}(t) < \theta_{\mathrm{perc}}$ holds for L consecutive frames. In this visualization, $L = 1$ and $\theta_{\mathrm{occ}} = \theta_{\mathrm{perc}} = 0.8$. The next segment begins with a one-frame overlap, and the procedure repeats.

et al., 2025), which estimates the dense tracking field $\mathbf{M}$ from the segment's initial frame to all subsequent frames; for each pixel in the initial frame, it provides a trajectory specifying its corresponding location at each time step, together with per-frame visibility and confidence. AllTracker's reliance on its occlusion-aware, geometry-sensitive formulation yields trajectories that remain stable under large motions and viewpoint changes. This is precisely the behavior required by our diffusion decoder.

4.3 Content-Adaptive Keyframe Selection Implementation

In the encoder, we implement content-adaptive keyframe selection by computing occupancy and perceptual similarity metrics between the original video frames and the forward-splatted rendering of the keyframe for each potential segment endpoint. Given the dense tracking field $\mathbf{M}$ from the current keyframe, we forward-splat the keyframe $x_1^{(k)}$ along the displacements $\{\mathbf{u}_t(p)\}_{p\in\Omega}$ to obtain splatted images $\tilde{x}_t^{(k)}$ and binary occupancy masks:

$$O_t(q) = \mathbb{I}\Big[\,\exists\, p \in \Omega :\ p + \mathbf{u}_t(p) = q\,\Big]. \tag{6}$$

The occupancy fraction at time t is

$$\mathrm{occ}(t) \;=\; \frac{1}{|\Omega|}\sum_{q\in\Omega} O_t(q), \tag{7}$$

measuring what fraction of the target frame is covered by the forward-warped keyframe. For perceptual similarity, we compute

$$\mathrm{sim}_{\mathrm{perc}}(t) \;=\; 1 - \mathrm{LPIPS}\big(O_t \odot \tilde{x}_t^{(k)},\, O_t \odot x_t^{(k)}\big), \tag{8}$$

where $\odot$ denotes element-wise multiplication and LPIPS (Zhang et al., 2018) is a learned perceptual similarity metric. Masking both images with O_t restricts LPIPS to covered regions, avoiding penalties from uncovered areas. We then define the combined keyframe-validity score $\theta_t \;:=\; \min\left\{\frac{\mathrm{occ}(t)}{\theta_{\mathrm{occ}}}, \frac{\mathrm{sim}_{\mathrm{perc}}(t)}{\theta_{\mathrm{perc}}}\right\}$, where θ_{occ} and θ_{perc} are the occupancy and perceptual-similarity thresholds, respectively. We define the endpoint of segment k as the earliest frame index $t > \tau_{k-1}$ such that this combined score remains below one for L consecutive frames, i.e.,

$$\tau_k := \min_{t>\tau_{k-1}} \Big\{ t \;\Big|\; \theta_{t+\ell} < 1 \;\; \forall\, \ell \in \{0,\dots,L-1\} \Big\}. \tag{9}$$

In other words, τ_k is the first frame in the earliest run of L consecutive frames for which the keyframe no longer provides sufficient coverage or perceptual similarity. The parameter L acts as a hysteresis term,

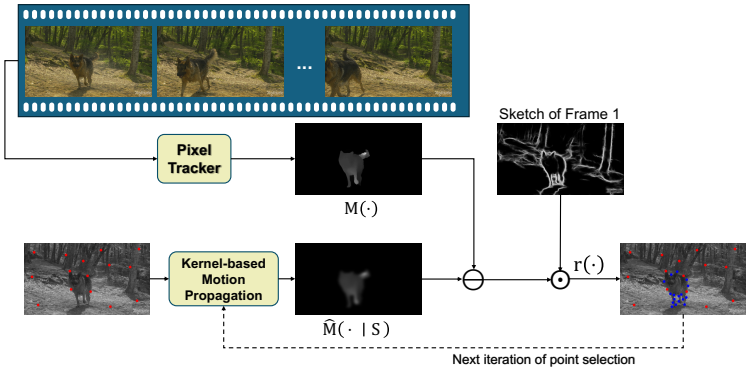

Figure 3: **Budget-Aware Sparse Trajectory Selection.** Given the current tracking set S (red dots), the dense tracking field is estimated as $\widehat{\mathbf{M}}(\cdot \mid S)$ using the RBF kernel interpolation in equation 3. The residual $r(p)$, as defined in equation 11, quantifies the discrepancy between the dense tracking field $\mathbf{M}(\cdot)$ and its reconstruction $\widehat{\mathbf{M}}(\cdot \mid S)$. Points with the largest sketch-weighted residuals (blue dots) are added to S, improving motion reconstruction in underrepresented or high-error regions. This procedure repeats until the bit-rate budget is reached or the reconstruction error converges.

preventing isolated score drops from triggering a new segment boundary. The corresponding segment length is $T_k = \tau_k - \tau_{k-1} + 1$. Figure 2 illustrates this process. Once a keyframe boundary is identified, the next segment begins with a one-frame overlap, and the procedure repeats. This implementation ensures that segment lengths adapt to scene content: static scenes yield longer segments with fewer keyframes, while dynamic scenes with occlusions or shot changes yield shorter segments with more frequent keyframes, thereby balancing keyframe coding cost against reconstruction quality.

4.4 Sparse Trajectory Conditioning Implementation

While the dense tracking field $\mathbf{M}$ provides complete motion information, transmitting all pixel trajectories is prohibitively expensive. We therefore introduce a budget-aware point selection strategy that identifies a sparse subset $S \subset \Omega$ of maximally informative tracks. We seek a subset $S \subset \Omega$ with $|S| \leq B$ that approximately minimizes the weighted reconstruction objective $\mathcal{R}(S)$ in Eq. equation 4. We adopt a budget-aware greedy selection that iteratively reduces $\mathcal{R}(S)$. To define the importance weights $w^{(k)}$, we apply Holistically-Nested Edge Detection (HED) (Xie & Tu, 2015) to the segment's first frame, yielding an edge map, which we refer to as the sketch of the frame:

$$w^{(k)}(p) \;=\; \mathrm{HED}\Big(x_1^{(k)}\Big)(p), \quad p \in \Omega. \tag{10}$$

Initialization. We partition the image into a coarse grid and select the pixel with maximum sketch strength in each cell, ensuring initial spatial coverage of the tracking set.
RBF Kernel Interpolation. Given a current set S of selected points, we estimate the dense displacement field $\widehat{\mathbf{M}}(\cdot \mid S)$ using the RBF kernel in equation 3. Specifically, the interpolation weights are determined by the Euclidean distance between pixels, so that nearby selected points have greater influence on the estimated motion field. Starting from the sketch-initialized grid S_0, we select the kernel bandwidth σ by minimizing the weighted reconstruction objective $\mathcal{R}(S_0)$ in equation 4, and use this σ for subsequent refinement of S.
Greedy Refinement. We refine S iteratively by adding points at which the current reconstruction exhibits the largest residual. At iteration m, we compute a residual map

$$r(p) \;=\; \frac{1}{T_k}\sum_{t=1}^{T_k} w^{(k)}(p)\,\big\|\mathbf{M}_t(p) - \widehat{\mathbf{M}}_t(p \mid S_m)\big\|_2^2, \tag{11}$$

identify local maxima of $r(\cdot)$ over Ω, and add the top candidates to form S_{m+1}. We then recompute $\widehat{\mathbf{M}}_t(\cdot \mid S_{m+1})$ and iterate. The process terminates when the budget is exhausted ($|S_m| = B$) or when $r(p)$

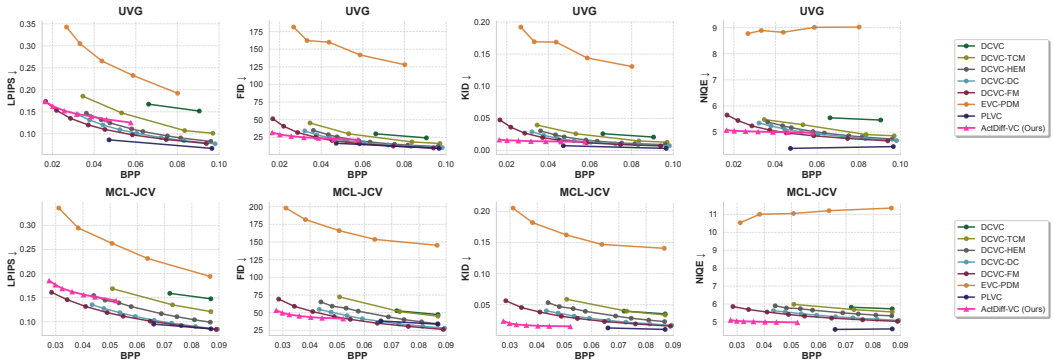

Figure 4: Quantitative comparison on the UVG and MCL-JCV datasets. We report LPIPS, FID, KID, and NIQE as a function of bits per pixel (BPP); lower values indicate better performance for all metrics.

falls below a tolerance for all $p \in \Omega$. Figure 3 illustrates this iterative refinement. Upon termination of the iterative refinement, the resulting sparse trajectory set $\mathcal{P}^{(k)}$, as defined in Eq. equation 2, provides reliable tracking, strong spatial coverage, and high informativeness under the fixed budget constraint.

4.5 Entropy Coding of Components

We entropy-code the keyframes and sparse trajectory set to remove statistical redundancy. Keyframes $x_1^{(k)}$ and $x_{T_k}^{(k)}$ are compressed using HiFiC (Mentzer et al., 2020). For the sparse trajectory set $\mathcal{P}^{(k)}$, each selected point $q \in S$ is represented by its initial location q together with temporal displacement differences $\Delta \mathbf{u}_t(q) = \mathbf{u}_t(q) - \mathbf{u}_{t-1}(q)$ for $t = 2, \ldots, T_k$. The sequence $\{\Delta \mathbf{u}_t(q)\}$ is then jointly entropy coded using a learned Huffman code derived from the empirical symbol distribution. The Huffman codebook is transmitted as part of the bitstream, and its overhead is included in the reported bitrate. Because motion deltas exhibit temporal correlation, this coding achieves significant compression. The initial points $\{q : q \in S\}$ and the segment length T_k are transmitted as side information without additional entropy coding.

5 Evaluation

5.1 Experimental Setup and Evaluation Metrics

We evaluate our approach on the UVG (Mercat et al., 2020) and MCL-JCV (Wang et al., 2016) datasets. We report perceptual quality using LPIPS (Zhang et al., 2018), FID (Heusel et al., 2017), KID (Bińkowski et al., 2018), and NIQE (Mittal et al., 2012), which are standard metrics for evaluating generative reconstruction. We focus on perceptual metrics rather than distortion-based measures such as PSNR, as our goal is to assess visual realism in the ultra-low-bitrate regime where pixel-wise fidelity is less informative.

Compared Methods. We compare our method against several learned video compression baselines, including DCVC (Li et al., 2021), DCVC-TCM (Sheng et al., 2022), DCVC-HEM (Li et al., 2022), DCVC-DC (Li et al., 2023), and DCVC-FM (Li et al., 2024b), as well as the GAN-based method PLVC (Yang et al., 2022) and the diffusion-based method EVC-PDM (Li et al., 2024a)[1]. We evaluate all competing methods using their official codebases on both datasets at a resolution of 480×720 whenever supported by the released implementations. The only exception is EVC-PDM, which we evaluate at 128×128 because its released implementation only supports that input resolution.

5.2 Experimental Results

Quantitative Comparison. Figure 4 compares ActDiff-VC with existing methods on the UVG and MCL-JCV datasets in terms of LPIPS, FID, KID, and NIQE (lower is better for all metrics). We focus on perceptual metrics, as distortion metrics such as PSNR are less informative in the ultra-low-bitrate setting.

[1] We compare against these methods as they represent the most relevant prior work with publicly available code and pretrained weights, enabling fair and reproducible evaluation.

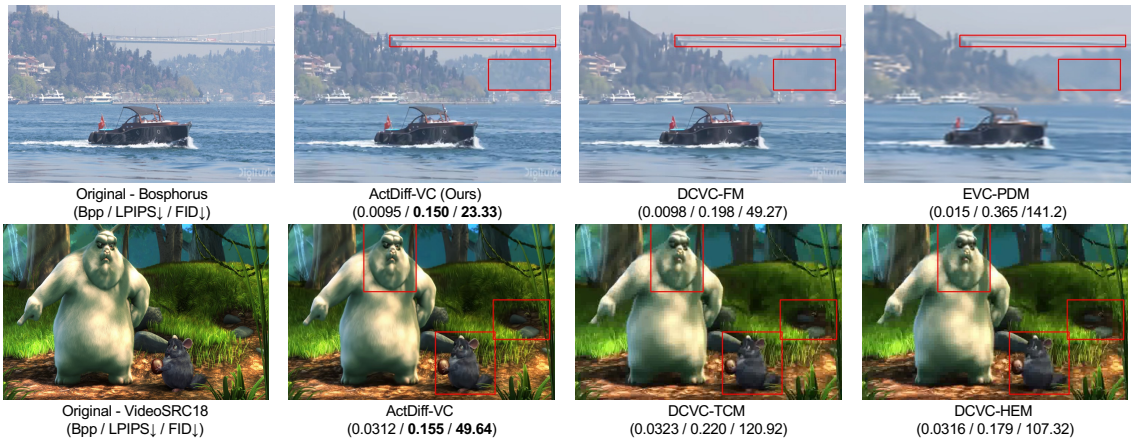

Figure 5: Qualitative comparison on representative sequences from UVG and MCL-JCV. We compare ActDiff-VC against competing methods at comparable, and in some cases higher, bit rates. The numbers below each reconstruction report per-video BPP, LPIPS, and FID (lower is better). Red boxes highlight challenging regions where competing methods exhibit stronger blur, artifacts, or structural degradation, while ActDiff-VC better preserves fine details, textures, and boundaries.

ActDiff-VC shows its clearest advantage in NIQE and KID in the ultra-low-bitrate regime. In terms of NIQE, ActDiff-VC reaches 5.08 at 0.0164 BPP on UVG and 5.10 at 0.0276 BPP on MCL-JCV, whereas DCVC-FM requires approximately 0.0372 BPP and 0.0780 BPP, respectively, to achieve comparable NIQE values. This corresponds to bitrate reductions of 55.9% on UVG and 64.6% on MCL-JCV. A similar trend is observed for KID: at nearly identical low bit rates, ActDiff-VC achieves 0.0168 at 0.0164 BPP on UVG and 0.0236 at 0.0276 BPP on MCL-JCV, improving over DCVC-FM by 64.6% and 58.0%, respectively.

FID also shows consistent gains in the ultra-low-rate regime. On UVG, ActDiff-VC obtains an FID of 31.94 at 0.0164 BPP, improving over DCVC-FM by 37.7% at a similar bitrate, while on MCL-JCV it achieves an FID of 53.40 at 0.0276 BPP, improving by 22.8% at a comparable rate. These results indicate that ActDiff-VC consistently improves distributional and no-reference perceptual metrics such as FID, KID, and NIQE at extremely low bitrates.

For LPIPS, ActDiff-VC is not the best overall but remains competitive in the low-rate regime. On UVG, at 0.0460 BPP, ActDiff-VC achieves an LPIPS of 0.1328, outperforming DCVC (0.1674 at 0.0662 BPP) and DCVC-TCM (0.1475 at 0.0532 BPP), while remaining higher than DCVC-FM (0.0976 at 0.0584 BPP). A similar pattern is observed on MCL-JCV, where ActDiff-VC achieves 0.1516 at 0.0444 BPP, outperforming DCVC (0.1589 at 0.0719 BPP) and the diffusion-based baseline EVC-PDM at comparable bit rates, although it remains behind DCVC-FM.

Compared with the GAN-based PLVC, which operates primarily at higher bit rates, ActDiff-VC extends perceptual compression to a substantially lower bitrate regime while maintaining similar trends in FID, KID, and NIQE. Compared with the diffusion-based baseline EVC-PDM, ActDiff-VC achieves substantial improvements across all four metrics on both datasets; at approximately 0.032–0.036 BPP, our method improves LPIPS by about 49%, FID by about 80%, KID by 91%, and NIQE by 49%.

Qualitative Results. Figure 5 shows qualitative comparisons on representative sequences from UVG and MCL-JCV. Across the selected examples, ActDiff-VC produces reconstructions with sharper local structure and fewer perceptual artifacts than competing methods, particularly in regions containing fine textures, boundaries, and small details. These examples are consistent with the quantitative results, showing that our method maintains perceptually realistic reconstruction even at very low bit rates. Additional qualitative examples are provided in Figure A2.

Keyframe Selection Analysis. Figure 6 illustrates the effect of our content-adaptive keyframe selection on videoSRC20 from MCL-JCV, where a scene cut occurs between Frame 42 and Frame 43. ActDiff-VC correctly identifies this transition and inserts a new keyframe, yielding a faithful reconstruction of the new scene immediately after the cut. In contrast, PLVC uses a fixed keyframe interval and cannot adapt to the

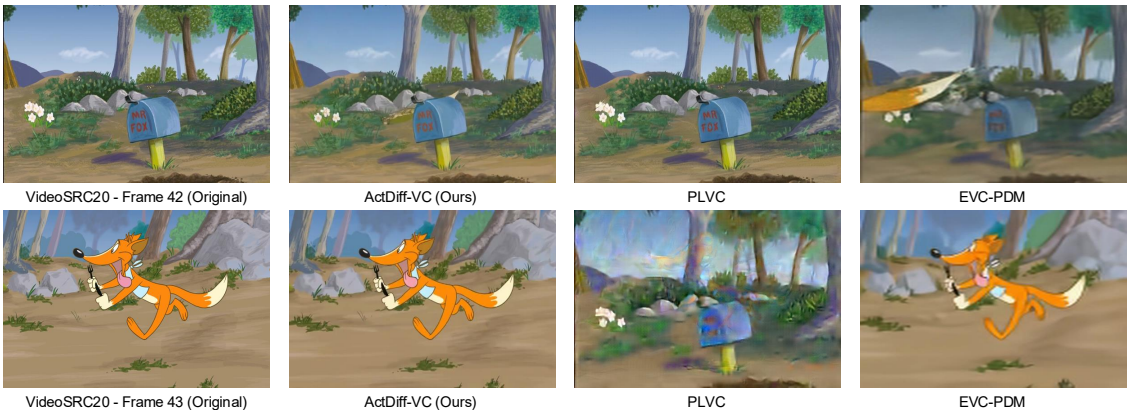

Figure 6: Visualization of content-adaptive keyframe selection. We show a frame before a scene change and the first frame after the cut, together with reconstructions from ActDiff-VC, PLVC, and EVC-PDM. ActDiff-VC detects the transition and inserts a new keyframe, producing a high-quality reconstruction after the cut. In contrast, PLVC relies on a fixed keyframe interval and fails to adapt to the scene cut, so the reconstructed frame after the cut still resembles the previous scene and contains visible artifacts. EVC-PDM can also adapt to scene changes via diffusion-based forecasting at the encoder, but still produces lower visual quality than ActDiff-VC.

Components			Metrics (vs. baseline)	
Keyframe policy	Point selection strategy	Bidirectional cond.	ΔLPIPS ↓	ΔFID ↓
Adaptive GOP (baseline block)				
Adaptive	Content-aware (sketch-weighted)	✓	0.0000	0.0000
Adaptive	Content-aware (sketch-weighted)	✗	+0.0321	**+1.3356**
Adaptive	Uniform grid	✓	+0.0246	+5.6126
Adaptive	High-mag flow	✓	+0.0458	+12.4746
Adaptive	Content-aware (no sketch)	✓	**+0.0219**	+1.4193
Fixed GOP (comparison block)				
Fixed	Content-aware (no sketch)	✓	**+0.1323**	**+24.7450**
Fixed	Uniform grid	✓	+0.1336	+37.9850
Fixed	High-mag flow	✓	+0.1412	+41.2817
Fixed	Uniform grid	✗	+0.1764	+59.4039

Table 1: Ablations at fixed bitrate. $\Delta > 0$ indicates degradation vs. the baseline (Adaptive GOP + Content-aware (sketch-weighted) + bidirectional conditioning); lower is better. Best non-baseline ablation values within each GOP block are **bold**.

abrupt content change, causing the reconstructed frame after the cut to resemble the previous scene with visible artifacts and distortions. EVC-PDM can also respond to scene changes, but does so through diffusion-based forecasting at the encoder while still producing lower visual quality than ActDiff-VC. This example shows that content-adaptive keyframe selection is effective for handling abrupt temporal discontinuities while preserving reconstruction quality. Additional examples are provided in Figure A3 and Figure A4.

5.3 Ablation Study

Component Ablations. Table 1 reports ablations of three key components on UVG at a fixed bitrate: (i) keyframe policy (Fixed GOP vs. Adaptive GOP), (ii) trajectory selection strategy, and (iii) bidirectional conditioning. We report metric deltas relative to a baseline configuration (Adaptive GOP + content-aware (sketch-weighted) trajectory selection + bidirectional conditioning), where $\Delta > 0$ indicates degradation and lower is better. For the Fixed GOP setting, the keyframe interval is set to the mean segment length produced by Adaptive GOP. Overall, Adaptive GOP and content-aware trajectory selection provide the largest gains, while bidirectional conditioning further improves reconstruction quality.

Effect of bidirectional conditioning. Bidirectional conditioning consistently improves visual quality across all settings. Under a Fixed GOP with uniform sampling, adding bidirectional conditioning reduces ΔFID

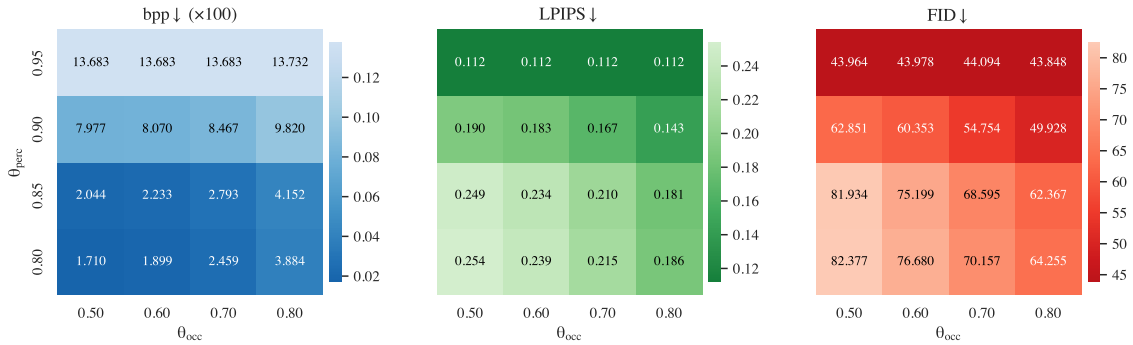

Figure 7: Sensitivity of Adaptive GOP thresholds. Heatmaps over occupancy threshold θ_{occ} (x-axis) and perceptual threshold θ_{perc} (y-axis), showing bitrate (bpp, ×100), LPIPS, and FID (lower is better). Increasing either threshold improves perceptual quality at the cost of higher bitrate.

from +59.4039 to +37.9850 and ΔLPIPS from +0.1764 to +0.1336, indicating its importance for stabilizing generative reconstruction.

Effect of keyframe policy. Replacing a Fixed GOP with Adaptive GOP substantially reduces degradation in both LPIPS and FID under matched settings. For example, under a uniform grid with bidirectional conditioning, ΔFID falls from +37.9850 (Fixed GOP) to +5.6126 (Adaptive GOP), while ΔLPIPS decreases from +0.1336 to +0.0246.

Effect of trajectory selection. Trajectory selection plays a key role in determining reconstruction quality. High-magnitude flow sampling over-focuses on large displacements while undersampling low-motion regions that are important for preserving texture, leading to worse LPIPS and FID compared to uniform sampling (e.g., ΔFID +12.4746 vs. +5.6126 under Adaptive GOP with bidirectional conditioning). Uniform sampling is more balanced but content-agnostic, as it does not account for motion reconstruction difficulty. In contrast, the proposed content-aware (sketch-weighted) selection achieves the best performance by prioritizing informative regions. Removing sketch weighting leads to small but consistent degradation (ΔFID +1.4193, ΔLPIPS +0.0219), showing that incorporating structural priors improves robustness. Without this weighting, displacement noise in low-texture regions propagates more easily and introduces visible artifacts.

Hyperparameter sensitivity for Adaptive GOP. Figure 7 shows the sensitivity of the Adaptive GOP procedure to its two termination thresholds, θ_{occ} and θ_{perc}, on high-motion MCL-JCV sequences with scene cuts. We evaluate θ_{perc} in a high-value range, as lower values rarely trigger keyframe insertion and therefore have limited impact on GOP segmentation. Both thresholds control the trade-off between bitrate and perceptual quality: increasing either threshold results in more conservative segment termination, yielding shorter GOPs (more frequent keyframes), higher bitrate, and improved reconstruction quality. Overall, θ_{occ} has a stronger and more consistent effect on the rate–quality trade-off. Increasing θ_{occ} leads to steady increases in bitrate and corresponding improvements in LPIPS and FID across all values of θ_{perc}. In contrast, θ_{perc} has a more pronounced effect only in the high-threshold regime, where increasing it from 0.85 to 0.90 or higher causes a sharp increase in bitrate together with significant gains in perceptual quality. Given our target operating regime of ultra-low bitrate compression (bpp ≤ 0.05), we select $\theta_{\text{perc}} = 0.85$ and $\theta_{\text{occ}} = 0.8$ in all experiments. This setting achieves 0.0415 bpp while maintaining strong perceptual quality, providing a favorable trade-off between compression efficiency and reconstruction fidelity.

6 Conclusion

In this work, we introduced **ActDiff-VC**, a diffusion-based video compression framework designed for the ultra-low-bitrate regime. Our approach combines a pre-trained conditional diffusion decoder with an active sampling strategy that selects sparse and informative conditioning signals for reconstruction. This design enables strong perceptual reconstruction quality at extreme compression rates. Experiments on standard benchmarks demonstrate the effectiveness of **ActDiff-VC** in this challenging setting, and ablation studies confirm the importance of its main components. Overall, our results suggest that diffusion-based generative models offer a promising direction for perceptual video compression in the ultra-low-bitrate regime.

References

David Alexandre, Hsueh-Ming Hang, and Wen-Hsiao Peng. Hierarchical b-frame video coding using two-layer canf without motion coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10249–10258, 2023.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

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

Marlene Careil, Matthew J Muckley, Jakob Verbeek, and Stéphane Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *The Twelfth International Conference on Learning Representations*, 2023.

Bolin Chen, Shanzhi Yin, Hanwei Zhu, Lingyu Zhu, Zihan Zhang, Jie Chen, Ru-Ling Liao, Shiqi Wang, and Yan Ye. Compressing human body video with interactive semantics: A generative approach. *arXiv preprint arXiv:2505.16152*, 2025.

Pengli Du, Ying Liu, Nam Ling, Yongxiong Ren, and Lingzhi Liu. Generative video compression with a transformer-based discriminator. In *2022 Picture Coding Symposium (PCS)*, pp. 349–353. IEEE, 2022.

Pengli Du, Ying Liu, and Nam Ling. Cgvc-t: Contextual generative video compression with transformers. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 14(2):209–223, 2024.

Xingtong Ge, Jixiang Luo, Xinjie Zhang, Tongda Xu, Guo Lu, Dailan He, Jing Geng, Yan Wang, Jun Zhang, and Hongwei Qin. Task-aware encoder control for deep video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26036–26045, 2024.

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2025.

Adam W Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Pavel Tokmakov, et al. Alltracker: Efficient dense point tracking at high resolution. *arXiv preprint arXiv:2506.07310*, 2025.

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. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Goluck Konuko, Giuseppe Valenzise, and Stéphane Lathuilière. Ultra-low bitrate video conferencing using deep image animation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4210–4214. IEEE, 2021.

Eric Lei, Yiğit Berkay Uslu, Hamed Hassani, and Shirin Saeedi Bidokhti. Text+ sketch: Image compression at ultra low rates. *arXiv preprint arXiv:2307.01944*, 2023.

Bohan Li, Yiming Liu, Xueyan Niu, Bo Bait, Wei Han, Lei Deng, and Deniz Gunduz. Extreme video compression with prediction using pre-trained diffusion models. In *2024 16th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1449–1455. IEEE, 2024a.

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

Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1503–1511, 2022.

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*, pp. 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*, pp. 26099–26108, 2024b.

Meiqin Liu, Chenming Xu, Yukai Gu, Chao Yao, and Yao Zhao. I^2vc: A unified framework for intra- & inter-frame video compression, 2024.

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*, pp. 11006–11015, 2019.

Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020.

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*, pp. 297–302, 2020.

Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

Michal Neoral, Jonáš Šerỳch, and Jiří Matas. Mft: Long-term tracking of every pixel. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6837–6847, 2024.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Maxime Oquab, Pierre Stock, Daniel Haziza, Tao Xu, Peizhao Zhang, Onur Celebi, Yana Hasson, Patrick Labatut, Bobo Bose-Kolanu, Thibault Peyronel, et al. Low bandwidth video-chat compression using deep generative models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2388–2397, 2021.

Zhihong Pan, Xin Zhou, and Hao Tian. Extreme generative image compression by learning text embedding from diffusion models. *arXiv preprint arXiv:2211.07793*, 2022.

Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. Lossy image compression with foundation diffusion models. In *European Conference on Computer Vision*, pp. 303–319. Springer, 2024.

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.

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.

Vijay Veerabadran, Reza Pourreza, Amirhossein Habibian, and Taco S Cohen. Adversarial distortion for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 168–169, 2020.

Rui Wan, Qi Zheng, and Yibo Fan. M3-cvc: Controllable video compression with multimodal generative models. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. Mcl-jcv: a jnd-based h. 264/avc video quality assessment dataset. In *2016 IEEE international conference on image processing (ICIP)*, pp. 1509–1513. IEEE, 2016.

Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 416–431, 2018.

Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

Chenming Xu, Meiqin Liu, Chao Yao, Weisi Lin, and Yao Zhao. Ibvc: Interpolation-driven b-frame video compression. *Pattern Recognition*, 153:110465, 2024.

Jiayu Yang, Yongqi Zhai, Wei Jiang, Chunhui Yang, Feng Gao, and Ronggang Wang. Adaptive prediction structure for learned video compression. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(2):1–23, 2024.

Ren Yang, Radu Timofte, and Luc Van Gool. Perceptual learned video compression with recurrent conditional gan. In *IJCAI*, pp. 1537–1544, 2022.

Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36:64971–64995, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

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*, pp. 586–595, 2018.

A1 Appendix Overview

This appendix provides additional discussion and analysis to complement the main paper. Section A2 discusses the main limitations of our method and directions for improving efficiency and robustness. Section A3 analyzes the encoding and decoding latency of the proposed method in comparison with competing codecs, highlighting the favorable encoder runtime of ActDiff-VC and the additional decoding cost introduced by diffusion-based reconstruction. Section A4 studies the effect of the sparse trajectory budget on compression performance, showing how the number of transmitted trajectories influences bitrate and perceptual quality. Section A5 describes the motion encoder construction used in our implementation: the transmitted sparse trajectory set is deterministically converted into a synthetic video representation, which is then encoded into the latent motion-conditioning input used by the diffusion decoder. Finally, Section A6 provides additional qualitative comparisons that extend the visual results in the main paper, together with further examples of the proposed content-adaptive keyframe selection under abrupt scene transitions.

A2 Limitations and Future Work

A primary limitation of our approach is the computational cost of diffusion-based decoding. Compared to conventional learned codecs, the iterative sampling process introduces higher latency and makes real-time deployment challenging. However, this trade-off is most relevant in the ultra-low-bitrate regime, where traditional codecs produce overly smooth or blurry reconstructions. In such settings, allocating additional computation at the decoder to achieve perceptually realistic reconstruction can be advantageous.

A key direction for future work is improving decoding efficiency. Recent advances in accelerated diffusion models, including model distillation and consistency models, offer promising avenues for reducing inference time while maintaining quality. Additionally, integrating motion estimation and compression into a unified end-to-end framework could further improve robustness and efficiency.

Our approach is particularly well-suited for scenarios where encoding must be fast and scalable, while decoding can be performed with higher computational budgets. Examples include one-to-many video streaming systems, where a central encoder distributes compressed content to many users, as well as cloud-based video delivery and archival storage, where perceptual quality at extremely low bitrates is critical.

A3 Encoding and Decoding Time Analysis

Time per frame (ms)	**DCVC**	**DCVC-TCM**	**DCVC-HEM**	**DCVC-DC**	**DCVC-FM**	**PLVC**	**ActDiff-VC (Ours)**
Encoding ↓	1244	353	360	347	140	40539	109
Decoding ↓	7137	129	81	126	116	141659	5528

Table A1: **Comparison of encoding and decoding time across competing methods.** We report the average encoding and decoding time per frame for all compared methods. For encoding time, we measure the runtime required to compress the input using each method and write the resulting bitstream to disk. For decoding time, we measure the runtime required to read the saved bitstream and reconstruct the frame. Lower is better for both metrics. ActDiff-VC achieves the fastest encoding time among all compared methods in this setting. Its decoding time is higher than conventional learned codecs, reflecting the cost of diffusion-based generation, but it remains substantially faster than the GAN-based PLVC.

Table A1 compares the average per-frame encoding and decoding latency of ActDiff-VC against competing methods.[2] All experiments were conducted on a single NVIDIA A100 GPU. For ActDiff-VC, encoding time includes dense tracking, content-adaptive keyframe selection, sparse trajectory selection, keyframe compression, and entropy coding of the transmitted side information. To measure encoding time, we run

[2] We do not include EVC-PDM in Table A1 because its released implementation only supports compression at a resolution of 128×128, whereas all other methods are evaluated at 480×720. Since this operating resolution is substantially smaller, its runtime is not directly comparable. For reference, EVC-PDM reports an average encoding time of 2308 ms per frame and decoding time of 1353 ms per frame.

each codec to compress the input frames and save the produced bitstream. To measure decoding time, we read the saved bitstream and time the reconstruction process at the decoder. The results show that ActDiff-VC has the fastest encoder among all compared methods, requiring only 109 ms per frame. This suggests that our compression pipeline is efficient on the encoder side despite relying on content-adaptive segmentation and sparse trajectory extraction. On the decoder side, ActDiff-VC is slower than conventional learned codecs, which is expected because reconstruction requires conditional diffusion-based generation rather than direct feed-forward decoding. Nevertheless, our decoder remains substantially faster than the GAN-based PLVC, whose decoding cost is much higher. Overall, these results indicate that ActDiff-VC offers a particularly favorable encoding-time profile, while its decoding latency reflects the computational cost of diffusion-based reconstruction.

A4 Impact of the Sparse Tracking Budget

Point budget	bpp ↓	LPIPS ↓	FID ↓
40	0.017949	0.1666	37.780
80	0.018640	0.1598	36.939
120	0.019312	0.1523	35.288
160	0.019977	0.1481	34.616
200	0.020633	0.1463	34.174
250	0.021441	0.1463	33.720
300	0.022233	0.1460	33.409
350	0.023020	0.1455	33.275
500	0.025023	0.1451	32.596

Table A2: **Effect of the point budget on the compression–quality trade-off.** We vary the maximum number of transmitted point trajectories per segment while keeping all other components fixed. We report bitrate (bpp) together with perceptual quality metrics (LPIPS and FID) aggregated over the evaluation set. Increasing the point budget mildly increases bitrate while generally improving perceptual quality, with diminishing returns beyond moderate budgets. Lower is better for bpp, LPIPS, and FID.

Our method represents motion side information using a sparse trajectory set $\mathcal{P}^{(k)}$ (Eq. equation 2), obtained by subsampling the dense tracking field $\mathbf{M}$. A key design parameter is the point budget B, which limits the maximum number of transmitted trajectories per GOP segment. To study its effect on compression performance, we sweep B on the UVG dataset while keeping all other components fixed.

Table A2 shows the resulting bitrate together with LPIPS and FID. As the point budget increases, bitrate rises gradually because the majority of the total bitrate is consumed by keyframe encoding, so increasing the number of transmitted trajectories contributes only a relatively small additional cost. At the same time, reconstruction quality generally improves, with the largest gains appearing at smaller budgets and progressively diminishing returns as B increases. This behavior is consistent with the strong spatial correlation of dense trajectories discussed in Section 3, which allows a relatively small subset of tracked points to provide informative motion conditioning for reconstruction. Based on this trade-off, we use a point budget of $B = 300$ in the main experiments.

A5 Motion Encoder Construction

This section describes how the sparse trajectory set $\mathcal{P}$ is converted into the motion-conditioning latent used by the diffusion decoder. In principle, sparse trajectories could be encoded in different ways, for example through a learned trajectory encoder or another structured motion representation. In our implementation, since the decoder is built on DaS, we use a synthetic video representation that matches the DaS conditioning format. The transmitted side information consists of the sparse trajectory set $\mathcal{P}$, and the decoder deterministically converts $\mathcal{P}$ into this synthetic video representation.

Figure A1: Visualization of the synthetic video representation used for motion conditioning. Given the sparse trajectory set $\mathcal{P}$, we generate a black-background RGB video in which each selected point is rendered as a small colored rectangle following its tracked trajectory. This synthetic video is then encoded by the pretrained VAE encoder $\mathcal{E}$ to obtain the tracking latent z_{trk} used by the diffusion decoder. The first row shows the generated synthetic video representation, while the second row shows the corresponding original video frames on which tracking is performed.

Let the sparse trajectory set for a segment of length T be

$$\mathcal{P} = \left\{ \left(q_i, \{u_t(q_i)\}_{t=1}^{T} \right) \right\}_{i=1}^{N},$$

where $q_i \in \Omega$ denotes the initial location of the i-th selected point in the first frame, and $u_t(q_i) \in \mathbb{R}^2$ denotes its displacement at time t. From this set, we construct a synthetic RGB video

$$V_{\mathcal{P}} \in [0,1]^{T \times H \times W \times 3},$$

initialized with a black background. For each tracked point q_i and each time step t, we compute its tracked position

$$p_{i,t} = q_i + u_t(q_i),$$

and render a small filled rectangle centered at $p_{i,t}$ on frame t. Each point is assigned a fixed color c_i determined by its first-frame location, so that the same trajectory can be visually followed across time. The synthetic video representation is then mapped to the latent space by the pretrained VAE encoder $\mathcal{E}$. Accordingly, the tracking latent used by the diffusion model is

$$z_{\text{trk}} = \mathcal{E}(V_{\mathcal{P}}) \in \mathbb{R}^{T_{\text{lat}} \times h \times w \times 16}.$$

Equivalently, the trajectory-conditioning pathway denoted by $\mathcal{E}_{\text{trk}}$ can be viewed as the composition

$$\mathcal{E}_{\text{trk}}(\mathcal{P}) = \mathcal{E}(V_{\mathcal{P}}),$$

where the conversion from $\mathcal{P}$ to $V_{\mathcal{P}}$ is deterministic. This notation emphasizes that $\mathcal{E}_{\text{trk}}$ is not an additional learned motion VAE used for compression; rather, it denotes the decoder-side procedure that transforms sparse trajectories into the latent motion-conditioning input required by the diffusion model.

Importantly, $V_{\mathcal{P}}$ is only an internal decoder representation. The bitstream contains the sparse trajectory set $\mathcal{P}$, not the synthetic video. Figure A1 illustrates this construction: the first row shows the synthetic video representation generated from the sparse trajectories, while the second row shows the corresponding original video frames on which tracking is performed.

A6 Additional Visualization

Figure A2 provides additional qualitative comparisons that extend the results in Figure 5. Across these additional examples, the same trends are consistently observed: ActDiff-VC produces cleaner reconstructions

Figure A2: Additional qualitative comparison examples extending Figure 5 on representative sequences from the UVG dataset (ShakeNDry) and the MCL-JCV dataset (videoSRC07 and videoSRC17). For each example, we compare ActDiff-VC with competing methods at similar or higher bit rates. The numbers below each reconstruction report per-video BPP, LPIPS, and FID (lower is better). Red boxes highlight regions where competing methods exhibit stronger artifacts, blur, or structural degradation, while ActDiff-VC better preserves visually plausible textures and object boundaries.

with fewer artifacts and better preservation of fine structural details than competing methods. Additional examples of our content-adaptive keyframe selection, extending Figure 6, are shown in Figure A3 and Figure A4. These examples further confirm the robustness of our method under abrupt scene transitions.

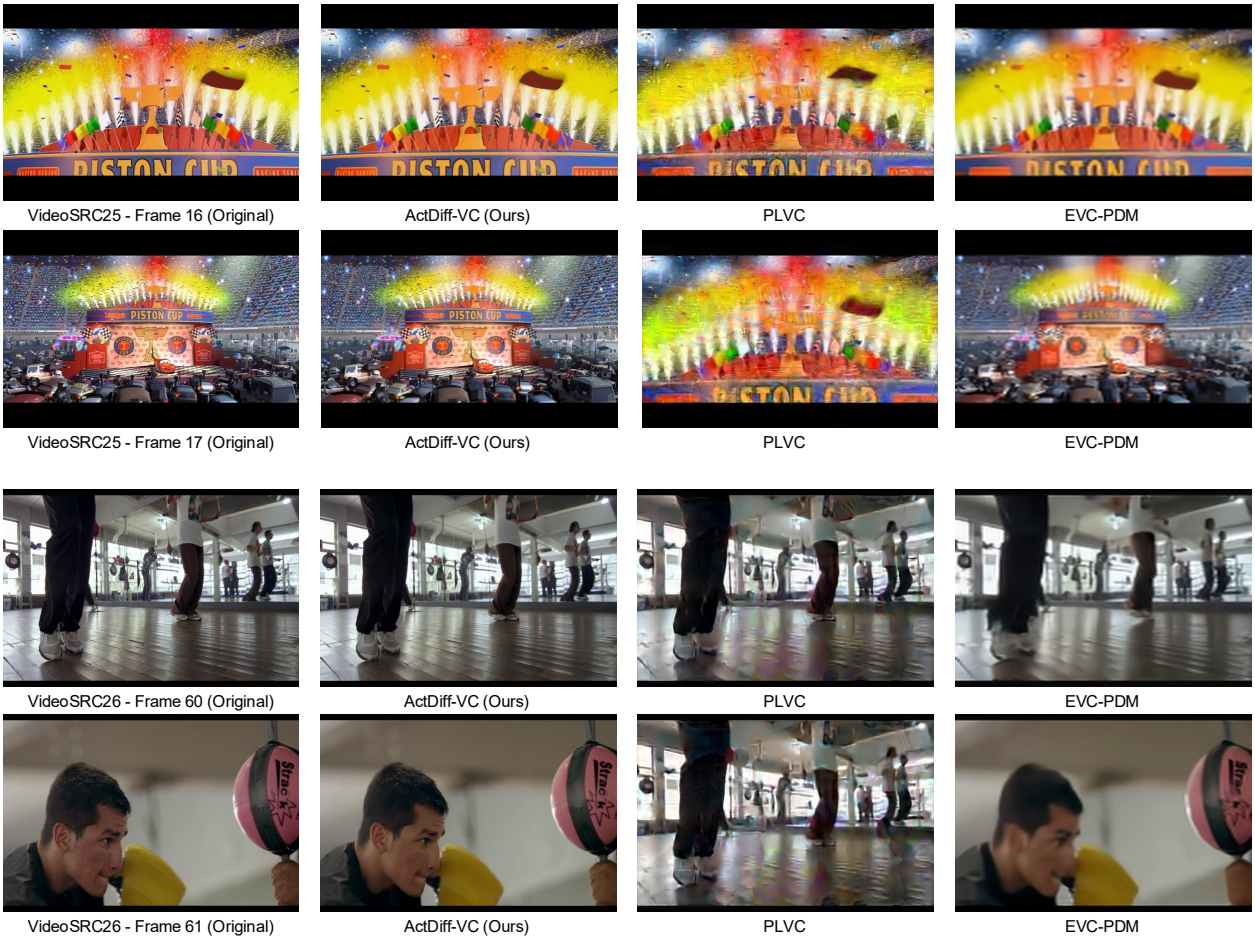

Figure A3: Additional visualizations of the proposed content-adaptive keyframe selection mechanism on MCL-JCV sequences videoSRC25 and videoSRC26. Similar to the examples in Figure 6, ActDiff-VC successfully adapts to abrupt scene changes by selecting new keyframes when needed, leading to higher-quality reconstructions after scene cuts. In contrast, PLVC often fails under its fixed key-frame schedule, while EVC-PDM requires substantially higher encoder-side computation and still produces weaker visual quality.

Figure A4: Additional visualizations of the proposed content-adaptive keyframe selection mechanism on MCL-JCV sequences videoSRC27 and videoSRC29. Similar to the examples in Figure 6, ActDiff-VC successfully adapts to abrupt scene changes by selecting new keyframes when needed, leading to higher-quality reconstructions after scene cuts. In contrast, PLVC often fails under its fixed key-frame schedule, while EVC-PDM requires substantially higher encoder-side computation and still produces weaker visual quality.