

Diffusion Autoencoders are Foundation Video Compressors

Niccolò Niccoli

University of Florence

niccolo.niccoli@unifi.it

Leonardo Galteri

Pegaso Telematic University

leonardo.galteri@unipegaso.it

Lorenzo Seidenari

University of Florence

lorenzo.seidenari@unifi.it

Abstract

Latent diffusion models rely on powerful pretrained Variational Autoencoders (VAEs) to project images into a compact latent space where diffusion sampling is both expressive and computationally efficient. We demonstrate that this same latent representation can be exploited to build a practical video compressor whose rate-distortion performance surpasses the industrial H.265/HEVC standard across a wide range of bitrates. The proposed pipeline first encodes a sparse set of keyframes with the latent-diffusion VAE, after which the resulting latent vectors are entropy-coded to remove residual spatial redundancy and drive the bitrate down. Starting from these decoded keyframes, each Group-of-Pictures is reconstructed entirely in latent space through a diffusion-based temporal interpolation process that synthesizes intermediate frames conditioned on their temporal context. To push performance further, we introduce two complementary fine-tuning strategies that freeze the decoder while learning frame-specific latent codes: a zero-shot configuration that operates with the off-the-shelf model and an adaptive configuration that refines both the latent codes and a lightweight subset of encoder parameters for every input video. Extensive experiments on standard benchmarks confirm that, under all tested conditions, our method consistently delivers higher objective quality and better perceptual fidelity than H.265 at comparable or lower bitrates.

1. Introduction

In recent years, video has become one of the most dominant forms of media, significantly transforming the landscape of entertainment, communication and education. The swift expansion of high-definition video content, fueled by the growing popularity of streaming services such as Netflix and YouTube, complemented by the extensive use of social media platforms like TikTok and Instagram, in addition to the widespread utilization of video conferencing applications such as Zoom and Teams, has resulted in a dramatic and exponential increase in the production of global

video data. Therefore, the vast volume of video data poses significant difficulties in terms of storage, transmission and processing capacity. This situation underscores the urgent need for efficient and effective video compression methods, as these techniques are now more crucial than ever before.

Traditional video compression standards, such as the extensively utilized codecs H.264, H.265, also known as High Efficiency Video Coding (HEVC), and the latest H.266, referred to as Versatile Video Coding (VVC), have been established as industry norms due to their extensive robustness and broad compatibility. Despite their wide acceptance and effectiveness in numerous applications, these coding algorithms encounter considerable challenges when operating under conditions of very low bitrates. Specifically, they often struggle to preserve satisfactory levels of visual quality when subjected to extreme compression. This decline in visual fidelity at reduced bitrates poses a critical constraint, especially for applications constrained by limited bandwidth, as well as for scenarios that demand significant efficiency in terms of storage capacity.

In response to the increasing need for efficient video compression, recent research efforts have delved into the development of generative video compression models that capitalize on the sophisticated generative potential inherent in deep learning techniques. Significant contributions in this area, including the studies documented in [27] and [31], have introduced promising alternatives to the conventional codecs currently in use today. These innovative generative models are notable for their considerable advances in achieving compression at low bitrates. Nonetheless, they encounter several critical challenges that remain unresolved. Among these challenges are the consistent preservation of temporal coherence across consecutive frames, assurance of stable visual quality during video playback, and the necessity for swift, computationally efficient processes for both compression and decompression. Furthermore, the inherent complexity involved in adeptly capturing temporal dynamics through generative methodologies continues to represent a substantial technical obstacle in the field.

In recent years, the progress made in the realm of image compression via neural networks has notably impacted

the video compression sector. Notably, studies leveraging diffusion models for image compression [5, 33] have revealed the impressive capability of neural networks to outperform traditional image compression techniques. Despite these advancements, video compression technology remains somewhat underdeveloped relative to image compression, attributable to the added layers of complexity inherent in video processing. These complexities include the necessity for maintaining temporal coherence, accurately estimating motion, and performing efficient frame interpolation, which are aspects that distinctly differentiate the challenges of video compression from those encountered in static image compression.

Motivated by the aforementioned challenges, combined with the remarkable achievements of diffusion-based methodologies in the realm of image compression, this study explores a novel paradigm for neural video compression. This new approach stands out from previously suggested methodologies due to its distinctive utility of existing pre-trained encoding systems. In particular, it exploits the power of pre-trained variational autoencoders (VAEs) that have been developed within the framework of Latent Diffusion Models, as detailed in [26]. The fundamental concept of our strategy involves the strategic repurposing of these powerful, pre-trained models to not only encode keyframes but also execute temporal interpolation.

To achieve intra-frame compression, we transform chosen keyframes into a reduced dimensional latent space derived from a pre-trained Variational Autoencoder (VAE). For inter-frame compression, we accomplish this by purposefully eliminating certain intermediate frames between keyframes and subsequently applying diffusion-based methods of temporal interpolation. These sophisticated interpolation techniques allow for the efficient reconstruction of the frames that were initially omitted, thereby markedly decreasing the total bitrate associated with the data.

The main achievements of our work can be categorized as follows:

- We show that Latent Diffusion Variational Autoencoders (VAEs) naturally function as potent zero-shot image compression tools.
- We construct a state-of-the-art neural video compression framework by fusing extensive pre-trained VAEs with advanced diffusion-based temporal interpolation methods, surpassing the compression efficiency of conventional encoders such as H.265 and H.266.
- We propose a unified learning strategy tailored for individual videos, which co-optimizes the latent feature representations and encoding parameters. This approach leads to significant advances in the balance of distortion against perception, exceeding the improvements offered by preliminary zero-shot methodologies.

The proposed model exhibits promising outcomes, especially when operating at reduced bitrate levels. It effectively surpasses traditional video encoding methods in both visual fidelity and efficiency of compression. These findings underscore the model’s proficiency in achieving elevated compression ratios while incurring only moderate perceptual deterioration. This characteristic renders the model especially advantageous for applications with stringent bandwidth constraints, including, but not limited to, mobile streaming, video conferencing, and solutions reliant on cloud-based data storage.

2. Related Works

2.1. Image Compression

The paradigm of learned transform coding can be thoroughly examined through a comprehensive variational-autoencoder (VAE) framework, where significant roles are assigned to various components: the analysis transform, which functions as the encoder, is responsible for approximating the posterior distribution; the entropy model is used to represent the prior imposed upon the latent variable space; and the synthesis transform, serving as the decoder, is tasked with parameterizing the conditional likelihood of the data. By re-envisioning traditional transform coding within this probabilistic framework, a pathway has been paved for the seamless incorporation of the extensive methodological advancements achieved in the field of modern VAE research into the domain of lossy image compression. An exemplary advancement in this area is evidenced by the work of Ballé et al. [1], who pioneered the shift from manually crafted linear transforms to nonlinear autoencoders. These autoencoders are capable of simultaneously learning both forward and inverse transformations, thereby endowing the networks with the ability to effectively model the complex, higher-order statistical dependencies commonly found in natural imagery, dependencies that are inadequately captured by conventional DCT-like bases.

Subsequent work has refined each component of the VAE. Hierarchical VAEs introduce multiple stochastic latent layers that progressively model global structure, mid-level textures, and fine details. Such hierarchies improve rate–distortion performance because high-level information acts as hyperpriors that tighten the entropy model for lower layers, reducing the number of bits required to describe them [1]. In parallel, attention-based encoders/decoders learn to focus computational resources on semantically or spatially complex regions, producing sharper reconstructions where they matter most while still allocating fewer bits to smooth areas [7].

Beyond pure rate–distortion optimization (RDO), perceptual quality has become a major design goal. Adversarially trained approaches inject a discriminator to penalize

unrealistic textures, which encourages the generator to hallucinate plausible high-frequency details that would otherwise be discarded at very low bitrates. Mentzer et al.[23] showed that this perceptual loss can be combined with a distortion loss to traverse the full spectrum between PSNR-oriented reconstructions and photorealistic ones. More recently, diffusion-based compressors[33] treat decompression as a conditional denoising process: a powerful, pre-trained diffusion model stochastically refines a noisy sample until it matches the compressed latents, thereby filling in missing information in a globally consistent manner.

Alternative generative backbones have also been proven to be effective. Duan et al.[10] integrate hierarchical VQ-VAE, in which the latent space is quantized by a codebook learned on several spatial scales. This discrete representation simplifies the entropy coding and yields crisp reconstructions. Careil et al.[5], on the other hand, compress each image into a compact set of text tokens and latent features, then leverage a diffusion-based image generator conditioned on those tokens to reconstruct remarkably realistic images at extremely low bitrates (tens of bytes). The common thread across these advances is the shift from purely analytical transforms to powerful data-driven priors that blend compression with generative modeling.

2.2. Video Compression

Efficient video compression is critical because video streams dominate global Internet traffic, accounting for more than 80% of downstream data in recent reports. Traditional hybrid codecs such as H.264/AVC, H.265/HEVC and the latest H.266/VVC achieve impressive compression by combining block-based motion compensation, transform coding of residuals, in-loop filtering, and sophisticated context-adaptive binary arithmetic coding (CABAC). Nevertheless, their reliance on handcrafted predictors and transforms can produce blocking, ringing and blurring artifacts when the bitrate budget is severely constrained.

In the domain of neural video compression (NVC), components of the traditional video compression approach, where individual segments can be substituted, are progressively being replaced. This substitution is achieved by employing neural networks that are optimized in an integrated manner specifically for RDO. Furthermore, with end-to-end neural models, it is possible to substitute the whole classical pipeline, allowing for an all-encompassing neural solution. Early systems substituted motion estimation with optical flow networks whose parameters are learned jointly with the rest of the codec, thereby reducing the mismatch between motion prediction and residual coding. DVC [22] and HDVC [14] illustrate this conditional-coding paradigm, wherein the distribution of residuals is modeled given the motion field and previously decoded frames, tightening entropy bounds. Li et al. [19] focus on compressing the feature

difference between current and predicted frames, leading to lower redundancy of information.

Two reference-frame strategies dominate neural codecs. P-frame compression exploits unidirectional prediction from the immediate past frame [6, 21, 22], while B-frame compression extends this to bidirectional prediction using past and future references [9, 30, 34]. The latter typically achieves higher efficiency, but requires additional latency. Accurate entropy modeling remains central: most works employ context-adaptive probability models, but their floating-point implementations can diverge across hardware. Tian et al. [29] solve this by discretizing latent data in codebook indices and transmitting only integers, guaranteeing bit-exact cross-platform decoding.

Recently, a paradigm shift has emerged: generative video compression. Instead of transmitting every frame, the encoder sends a sparse set of keyframes or compact conditioning tokens; a powerful generative model then synthesizes intermediate frames during playback. Wu et al.[31] invert keyframes into text embeddings using Stable Diffusion, interpolate those embeddings over time, and decode them into video, effectively turning text into a streaming representation. Li et al.[18] show that even lower rates can be achieved by selecting a tiny subset of reference frames and letting a frozen diffusion prior hallucinate the rest. Complementary work on video tokenizers such as COSMOS [8] discretizes video into space-time tokens that can be fed to large language-vision models, thus unifying compression and generative modeling pipelines.

2.3. Temporal Consistency

While individual frame quality is important, perceptually pleasing video demands temporal consistency: consecutive frames must transition smoothly without flickering, ghosting or jitter. Classical codecs address this via motion-compensated prediction, but generative approaches require explicit temporal modeling. Lu et al.[22] already reduce temporal redundancy by learning motion in the latent space, whereas bidirectional motion estimation and warping[9, 15, 20] generate a high-quality prediction that serves as a strong prior for the residual encoder.

Within diffusion-based generators, temporal coherence can be injected by architectural modifications. Several works add 3D convolutions or temporal attention on top of the standard 2D U-Net backbone so that features attend jointly across spatial and temporal dimensions [2, 3, 36]. Xu et al.[32] extend these ideas with a recurrent, flow-guided feature-propagation module that persistently carries information forward through time, stabilizing textures over long sequences.

Cross-frame attention provides another route: Khachatrian et al.[16] condition every frame on the first frame to preserve global appearance, while Zhao et al.[35] con-

catenate all frames into a single large 'image' so that self-attention discovers spatial correspondences over time. A complementary strategy, flow-guided latent propagation, explicitly warps latent features from nearby frames according to optical flow; merging these warped features with the current latent stabilizes motion and suppresses flicker. To further polish transitions, Zhao et al. [35] introduce an interleaved frame smoother that refines pixels in the RGB domain.

Finally, Wu et al. [31] propose a novel gradient-based prompt fitting technique. Starting from a random text prompt, they iteratively update prompt embeddings to minimize pixel-wise error with a ground-truth reference frame, effectively 'baking in' visual identity information. A temporal regularization loss encourages neighboring frames to share similar prompts, thereby suppressing frame-to-frame drift. This elegant optimization bridges the gap between symbolic conditioning and pixel-level coherence, yielding videos that are both semantically faithful and temporally stable.

3. Method

We present a novel framework for video compression that integrates the compression of latent keyframes with the interpolation of temporal frames. This approach successfully yields high-fidelity reconstructions while simultaneously achieving a reduction in bitrate. The proposed architecture, as shown in Fig. 1, comprises two main modules: (i) a pre-trained Variational Autoencoder (VAE) that has the responsibility of generating rich and compact latent representations from a subset of keyframes, and (ii) a temporal interpolation network that synthesizes the intermediate frames between reconstructed frames. The interaction between these two components not only improves compression efficiency, but also ensures that perceptual quality is preserved. Our architecture allows learning the representation on the encoder side as a form of ulterior compression, meaning that the encoder continues to refine its latent space during training while the decoder remains frozen, guaranteeing decodability at the receiver without any additional parameter transmission.

3.1. Keyframe Compression

The initial phase of the proposed compression pipeline is identified as the encoding stage. During this phase, an input video, represented as $V = \{x_0, x_1, \dots, x_n\}$, undergoes processing to obtain a subset of keyframes selected at uniform intervals, formulated as $\{x_t \mid t = 0, \delta, 2\delta, \dots, N\delta\}$. This method of sampling at regular intervals facilitates easier deployment and eliminates the necessity for costly procedures associated with scene-change detection, yet our framework can seamlessly integrate with dynamic keyframe selection approaches when needed. Subsequent to extrac-

tion, each specific keyframe x_t is input into a pre-trained VAE encoder E_θ . This encoder transforms the data from its high-dimensional pixel form into a significantly reduced-dimensional latent space, producing compact representations denoted as $y_t = E_\theta(x_t)$. The application of a VAE that originates from a generative-diffusion model is advantageous because of the model's extensive experience in learning highly expressive feature spaces. These spaces encompass a diverse array of visual textures and semantic elements, thereby investing the latent codes with robust generalization capabilities.

After the encoding stage, we proceed to further minimize the dimension of the resultant latent representations. This is achieved through the integration of a lossless entropy-coding technique known as Asymmetric Numeral Systems (ANS) [11]. ANS utilizes the probability distribution of individual latent symbols to form a compact, reversible bitstream. Notably, the algorithm attains compression ratios close to the theoretical optimum, all while maintaining computational efficiency. Importantly, the lossless nature of ANS ensures that the latent representation retains perfect fidelity after decompression, with no introduction of quantization error at this phase; thus, all information generated by the Variational Autoencoder (VAE) is wholly preserved.

3.2. Decoding and Interpolation

During the decoding phase, the initially compressed keyframes y_t undergo the decoding process through the VAE decoder D , producing the reconstructed forms denoted as $\hat{x}_t = D(y_t)$. The decoder D , being the architectural counterpart to the encoder E_θ , effectively maps the latent codes back into the image domain with high fidelity. To reconstruct the temporal progression of the original video footage, we use a specialized pre-trained framework referred to as AMT [20]. This network, referred to as I , is responsible for generating intervening frames situated between sequentially restored keyframes, namely $\hat{x}_t$ and $\hat{x}_{t+\delta}$. The AMT framework employs a sophisticated integration of multi-scale attention methodologies alongside motion-aware layering techniques to creatively project potential motion trajectories. This approach ensures that the synthesized frames maintain temporal coherence with respect to the adjacent reference frames.

The process of selecting keyframes is dictated by the distance parameter δ , which denotes the temporal interval between keyframes in the original video sequence. This parameter governs the trade-off between the compression ratio and the quality of reconstruction. When δ increases, the number of transmitted keyframes decreases, thus improving the compression ratio. However, this necessitates more reliance on the interpolation network to reconstruct a larger portion of the frames, thereby amplifying the probability of errors during the reconstruction phase. However, decreas-

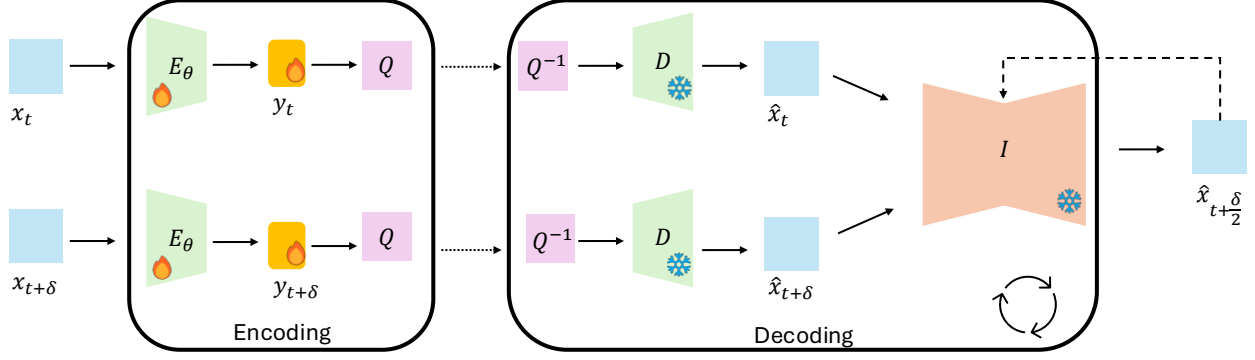


Figure 1. Diagram of the proposed compression framework. Encoders E_θ output latent vectors y_t for each input frame. After lossless compression and reconstruction by the quantizer Q , frames are decoded by D and temporal interpolation is performed by I iteratively to reconstruct all frames between x_t and $x_{t+\delta}$.

ing the value of δ enhances the quality of the reconstruction due to the shortened interpolation distance, although at the cost of a larger bitstream size. In practical applications, δ can be adaptively selected based on specific targets related to bitrate constraints or perceptual quality criteria.

The interpolation process is iterative and is applied to each pair of subsequent keyframes $(\hat{x}_t, \hat{x}_{t+\delta})$. First, we use I to synthesize the midpoint frame:

$$x_{t+\delta/2} = I(\hat{x}_t, \hat{x}_{t+\delta}),$$

which corresponds to the temporal location $t + \delta/2$. Subsequently, we continually implement the same interpolation methodology on the newly formed sub-intervals, specifically $[\hat{x}_t, x_{t+\delta/2}]$ and $[x_{t+\delta/2}, \hat{x}_{t+\delta}]$. This process iteratively refines our temporal resolution by progressively halving the intervals, ensuring that each intermediate frame between the original keyframes is meticulously reconstructed. The hierarchical approach effectively leverages the principle that smaller temporal discrepancies can usually be interpolated with higher precision, thus effectively mitigating interpolation errors. This strategy disperses potential inaccuracies more uniformly across the entire sequence, resulting in the generation of significantly smoother motion dynamics.

3.3. Encoder and Latents Fine-tuning

In order to adaptively master frame representation, we maintain the fixed weights of the VAE’s decoder and perform fine-tuning exclusively on the latent representations. Concurrently, we develop lightweight LoRA adapters for the VAE’s encoder. This carefully crafted design guarantees that the receiver can interpret any incoming compressed bitstream successfully without necessitating adjustments to the decoder’s model parameters. In addition, the encoder can easily adjust to a variety of content types, allowing optimal compression. Since we retain an unaltered decoder, there is the flexibility to fine-tune parameters for each individual

frame on its own, ensuring no compromise in decodability or compatibility with streams encoded earlier.

The loss function minimized during fine-tuning is

$$\mathcal{L}(x_t, \hat{x}_t) = \|x_t - \hat{x}_t\|_2 + \alpha \text{LPIPS}(x_t, \hat{x}_t), \quad (1)$$

where x_t denotes the originating keyframe, while $\hat{x}_t$ represents the keyframe that has undergone reconstruction following pipeline processing. Here, LPIPS indicates the metric for achieving learned perceptual similarity, which integrates both pixel-level and perceptual coherence. Through empirical evaluation, we determined the optimal setting for the scalar hyperparameter α to be 0.01. This specific selection facilitates an equilibrium between the precision of individual pixel values and the overall perceptual integrity. The process of acquiring the reconstructed keyframe $\hat{x}_t$ involves inputting the decompressed latent representation y_t into the pre-trained and immutable decoder D . This sequence ensures that $\hat{x}_t$ is faithfully reconstructed regarding both accuracy and perception.

All Variational Autoencoders (VAEs) exhibit a unified structural design, which is illustrated in Fig. 2. The encoder begins with an initial component known as a *DownBlock*. This stage incorporates a sequence of two-dimensional convolutional residual layers that are responsible for the progressive down-sampling of the input data. Following this, there is a *MidBlock*, which is composed of numerous residual blocks configured in a UNet-like structure. This configuration is characterized by skip connections and the incorporation of self-attention mechanisms. In the concluding segment, there exists a Group Normalization layer, succeeded by a two-dimensional convolutional layer. This terminal layer is designed to produce 16 output channels in the case of SD 3.5, whereas for the SDXL and SD 1.5 versions, it produces 4 output channels. These variations in the number of output channels signify the differences in the latent dimensional space among the various iterations of the model.

3.3.1. Low-Rank Adaptation of Encoder Parameters

To learn specialized representations for each keyframe while keeping the model lightweight, we employ Low-Rank Adapters (LoRA) [13], as illustrated in Fig. 2. We insert adapters at every layer of the *DownBlock* and at each layer of the *MidBlock*, leaving the remaining parameters frozen. Formally, for each block expressed as $B_i(x) = \Phi_i(x)$ we learn a low-rank adapter to compute

$$B_i(x) = \Phi_i(x) + \Delta_\Phi x,$$

where $\Delta_\Phi = BA$ with rank $r \ll d$ (the full dimension). This decomposition keeps the number of trainable parameters minimal, enabling per-video or even per-frame specialization without heavy memory overhead. First, we warm up the adapters by learning parameters in the whole video, then fine-tune each adapter on a single frame, using the parameters learned for frame t as initialization for frame subsequent $t + \delta$. Because all blocks of our model are differentiable, minimizing Eq. 1 directly updates the adapter weights.

3.3.2. Latent Fine-tuning

In addition to adapting the network weights, we also optimize each latent frame representation $y_t = E_\theta(x_t)$ through a lightweight procedure that involves only the encoder side. Given a latent vector $y_t^{(i)}$ in iteration i , we update it through a projected gradient descent:

$$y_t^{(i+1)} \leftarrow y_t^{(i)} + \eta \nabla_{y_t} \mathcal{L}(x_t, D(y_t^{(i)})), \quad (2)$$

where $\mathcal{L}$ is defined in Eq. 1, D is the frozen decoder, and η is the learning rate calculated using AdamW [17]. Because gradients propagate only through D , this refinement step effectively nudges the latent vector to regions of latent space that decode images with higher perceptual fidelity, without altering the decoder’s parameters. By iterating a handful of steps per frame, we observe noticeable improvements in sharpness and color consistency.

4. Experimental Results

4.1. Dataset and Protocol

We evaluated our method on two distinct video datasets. The first one is the UVG collection [24], which consists of sixteen high-quality videos covering a wide range of content types, including natural scenes, sports footage, and various human activities. Each UVG sequence is provided at 1920×1080 pixels and 120 fps, making it a widely accepted benchmark for research on low- and high-bitrate video coding. The second evaluation set is the curated subset introduced by Wu *et al.* [31]. It aggregates videos sourced from multiple public datasets, and therefore exposes our method to a diverse set of domains such as 2D

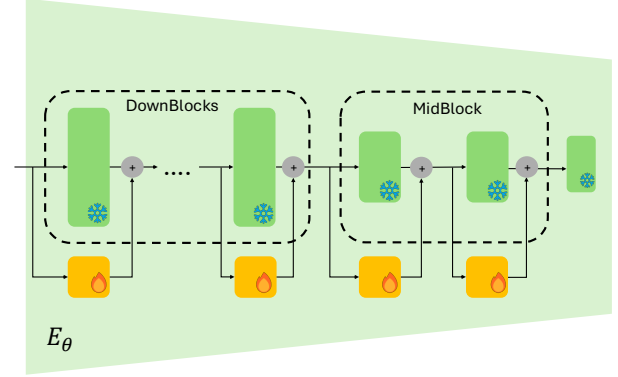


Figure 2. Detail of encoder fine-tuning. We attach a LoRA adapter to each *DownBlock* and to each layer of the *MidBlock*. The remaining parameters, shown in gray, are kept frozen during fine-tuning.

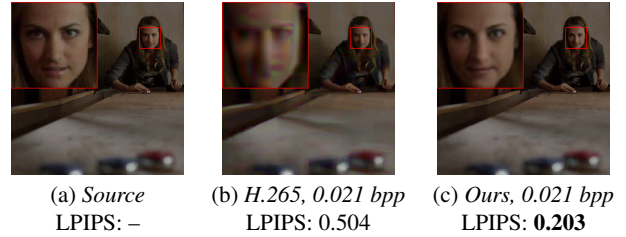


Figure 3. Qualitative comparison of Our approach (c) and H.265(a) at 0.021 bpp. We report the uncompressed frame (a) and the LPIPS ($\downarrow$).

animations, 3D gaming footage, and carefully filmed natural landscapes.

To ensure comparable results between the two datasets, all videos were resized and cropped to obtain a spatial resolution of 512×512 pixels. For sequences originally recorded at higher resolutions, we first applied bicubic down-scaling with anti-aliasing, and then cropped the central window. This pipeline preserves the most salient region of the frame while preventing border artifacts introduced by naïve down-sampling.

Unlike classical codecs that encode every frame, our framework can operate in a frame-sparse regime. Specifically, we modulate the extraction rate from the original video between: 1) encoding all frames (no interpolation), and 2) encoding one frame every $K \in \{2, 4, 8, 16, 32\}$, while iteratively interpolating the missing ones to reconstruct the full GOP. This strategy mimics the behavior of B-frames in traditional codecs but leverages neural interpolation to inpaint motion and fine details absent in the latent representation. The resulting flexibility is crucial for adapting the compression ratio to application-specific constraints.

4.2. Compression Performance

In our experiments, we employ various pre-trained Stable Diffusion encoders [12, 25, 26]. More precisely, we used the publicly available models in Hugging Face: SD 1.5¹, SD XL Turbo², and SD 3.5³.

It should be noted that the model that relies on the VAE shipped with SD 3.5 produces latent tensors of significantly higher dimensionality (16 channels as opposed to the 4 channels emitted by SD 1.5 and SD XL). Consequently, even when identical entropy models are used, the resulting bitstreams are several times larger. However, we include SD 3.5 in our study to showcase how richer representations trade storage cost for reconstruction fidelity. This comparison highlights the interaction between latent expressiveness, compression ratio, and training complexity, offering a more holistic perspective on the design space of latent diffusion video codecs.

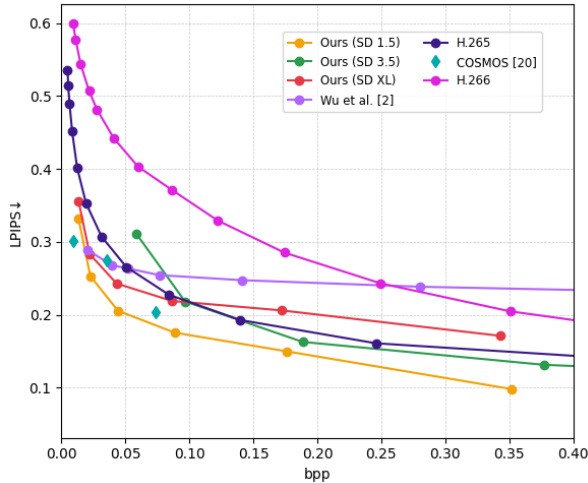


Figure 4. **Rate-distortion curves on the Wu *et al.* dataset [31].** The horizontal axis reports the average bits per pixel (bpp), whereas the vertical axis depicts LPIPS. Lower curves indicate better performance.

We tested the proposed method on both datasets [24, 31], reporting the corresponding rate-distortion curves in Fig. 4 (Wu *et al.*) and Fig. 5 (UVG). In both cases, we observe that, despite the reduced channel count in its latent space, the SD 1.5 encoder consistently achieves the lowest LPIPS across the entire spectrum of bit-rates. Therefore, it outperforms all competing baselines, including the industry standard H.265/HEVC [28], its successor H.266/VVC [4], and the recently released COSMOS tokenizer [8]. Interestingly,

¹<https://huggingface.co/stabilityai/sd-vae-ft-mse>

²<https://huggingface.co/stabilityai/sdxl-turbo>

³<https://stability.ai/news/introducing-stable-diffusion-3-5>

SD 3.5, which employs a substantially larger latent space, does not exceed SD 1.5 or SD XL below 0.5 bpp.

A qualitative illustration of this trend is provided in Fig. 3. At 0.021 bpp, the H.265 reconstruction exhibits visible ringing and block-like artifacts, while our SD 1.5 variant retains clean edges and richer textures, corroborating the numerical advantage.

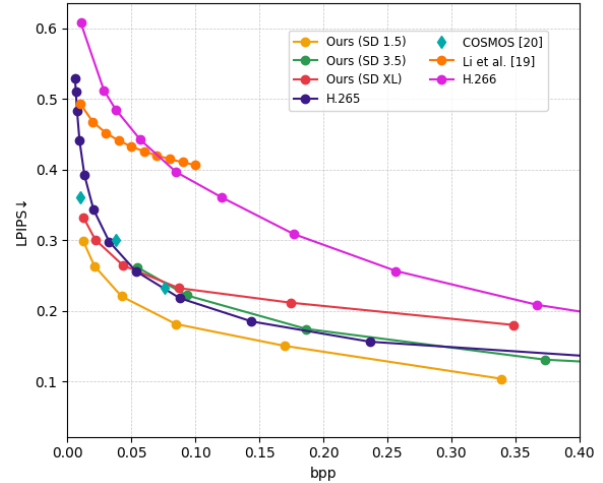


Figure 5. **Rate-distortion curves on the UVG dataset [24].** Our SD 1.5 codec traces the envelope of the graph, signalling a superior perceptual quality for any given bitrate.

In Fig. 4 we additionally include Promptus [31], a concurrent method that streams frame prompts instead of latent tensors. To the best of our knowledge, Promptus is the only low-training, latent diffusion-based approach directly targeting video compression. In Fig. 5, we show another competing method that leverages a neural compression network for keyframe compression and a diffusion model to generate frames from a selected subset of reference frames.

Finally, we benchmark against the mature, industry-grade codecs H.265 and H.266. Even under this rigorous comparison, our fine-tuned SD 1.5 model outperforms the anchors at all tested bitrates, while SD XL and SD 3.5 deliver competitive but slightly inferior curves.

4.3. Ablation Studies

Table 1 summarizes a controlled ablation study in which we investigate the relative importance of fine-tuning encoder parameters (VAE weights) and / or latent vectors. For each base model, we consider four configurations: (i) *zero-shot* (no additional training), (ii) *Encoder-ft* (VAE weights adapted via LoRA), (iii) *Latent-ft* (only the latent codes updated), and (iv) *full fine-tuning* (both components optimized jointly). Unsurprisingly, enabling both optimization paths yields the best LPIPS for every architecture. However, a more nuanced pattern emerges when the fine-

tuning budget is limited. For SD 1.5 and SD XL, adapting only the latent vectors provides a greater improvement than updating the encoder alone, suggesting that pre-trained weight priors already offer a sufficiently expressive basis. In contrast, SD 3.5 profits more from the adaptation of the encoder, arguably because its higher channel count makes latent space optimization more challenging.

Method	Encoder	Latent	LPIPS
SD 1.5	✗	✗	0.1389
	✓	✗	0.1198
	✗	✓	0.1149
	✓	✓	0.1112
SD XL	✗	✗	0.1999
	✓	✗	0.2118
	✗	✓	0.1915
	✓	✓	0.1895
SD 3.5	✗	✗	0.1995
	✓	✗	0.1953
	✗	✓	0.2135
	✓	✓	0.1953

Table 1. **Ablation study at 0.35 bpp** on the dataset of Wu *et al.* [31]. LPIPS scores are obtained by linear interpolation of our full rate-distortion curves (Fig. 4). Values in bold denote the best result within each block.

5. Conclusion

This work introduces a novel video compression approach that leverages the power of pre-trained Variational Autoencoders (VAEs) and Diffusion-based frame interpolation. By integrating these two paradigms, we capitalize on the representational efficiency of modern generative models and the temporal consistency offered by state-of-the-art interpolators, laying the groundwork for a next-generation codec that is both data-driven and hardware-friendly.

Our method efficiently compresses keyframes into a compact latent space and reconstructs intermediate frames through temporally aware diffusion interpolation, thereby reducing both spatial and temporal redundancy while maintaining, and in many cases enhancing, perceptual visual quality. In practice, this results in fewer transmitted bits, lower storage requirements, and graceful degradation under challenging network conditions, all without the need for content-specific retraining.

Our primary goal is to demonstrate the potential of this method to outperform traditional video codecs across a wide spectrum of resolutions and bitrates. Extensive experiments on benchmark datasets show consistent gains in perceptual quality, particularly at low-to-medium bitrates, where legacy codecs such as H.264/AVC and H.265/HEVC typically struggle. Zero-shot encoding performance can

be further enhanced by selectively fine-tuning latent and encoder parameters via low-rank adaptation (LoRA), a lightweight procedure that requires only a handful of gradient steps and can be executed on consumer-grade GPUs.

In conclusion, our proposed framework offers a promising and forward-compatible solution for video compression. Compared to other methods, it relies only on pre-trained components, making it both modular and easy to update in every part with a newer version. Looking ahead, we believe that coupling our approach with scalable transformer-based VAEs, perceptual loss functions, and hardware-accelerated inference will unlock further gains, ultimately bridging the gap between research prototypes and production-ready video delivery systems.

References

- [1] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 2
- [2] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, V. Jampani, and R. Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3
- [3] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, pages 22563–22575, 2023. 3
- [4] 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. 7
- [5] M. Careil, M. J. Muckley, J. Verbeek, and S. Lathuilière. Towards image compression with perfect realism at ultra-low bitrates. In *The Twelfth ICLR*, 2023. 2, 3
- [6] Z. Chen, T. He, X. Jin, and F. Wu. Learning for video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):566–576, 2019. 3
- [7] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *CVPR*, pages 7939–7948, 2020. 2
- [8] cosmos. Nvidia cosmos. <https://github.com/NVIDIA/Cosmos>, 2025. 3, 7
- [9] A. Djelouah, J. Campos, S. Schaub-Meyer, and C. Schroers. Neural inter-frame compression for video coding. In *ICCV*, pages 6421–6429, 2019. 3
- [10] Z. Duan, M. Lu, Z. Ma, and F. Zhu. Lossy image compression with quantized hierarchical vaes. In *WACV*, pages 198–207, 2023. 3
- [11] J. Duda. Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding. *arXiv preprint arXiv:1311.2540*, 2013. 4
- [12] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T.

- Dockhorn, E. Zion, K. Lacey, A. Goodwin, Y. Marek, and R. Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 7
- [13] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6
- [14] Y. Hu, C. Jung, Q. Qin, J. Han, Y. Liu, and M. Li. Hdvc: Deep video compression with hyperprior-based entropy coding. *IEEE Access*, 2024. 3
- [15] X. Jin, L. Wu, G. Shen, Y. Chen, J. Chen, J. Koo, and C. Hahm. Enhanced bi-directional motion estimation for video frame interpolation. In *WACV*, pages 5049–5057, 2023. 3
- [16] L. Khachatryan, A. Movsisyan, V. Tadevosyan, R. Henschel, Z. Wang, S. Navasardyan, and H. Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *ICCV*, pages 15954–15964, 2023. 3
- [17] Ilya L. and Frank H. Decoupled weight decay regularization. In *ICLR*, 2019. 6
- [18] B. Li, Y. Liu, X. Niu, B. Bai, L. Deng, and D. Gündüz. Extreme video compression with pre-trained diffusion models. *arXiv preprint arXiv:2402.08934*, 2024. 3
- [19] J. Li, B. Li, and Y. Lu. Neural video compression with feature modulation. In *CVPR*, pages 26099–26108, 2024. 3
- [20] Z. Li, Z. Zhu, L. Han, Q. Hou, C. Guo, and M. Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *CVPR*, pages 9801–9810, 2023. 3, 4
- [21] H. Liu, M. Lu, Z. Ma, F. Wang, Z. Xie, X. Cao, and Y. Wang. Neural video coding using multiscale motion compensation and spatiotemporal context model. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(8): 3182–3196, 2020. 3
- [22] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao. Dvc: An end-to-end deep video compression framework. In *CVPR*, pages 11006–11015, 2019. 3
- [23] F. Mentzer, G. Toderici, M. Tschannen, and E. Agustsson. High-fidelity generative image compression. *NeurIPS*, 33: 11913–11924, 2020. 3
- [24] A. Mercat, M. Viitanen, and J. Vanne. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 297–302, 2020. 6, 7
- [25] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 7
- [26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2, 7
- [27] M. Shukor, B. B. Damodaran, X. Yao, and P. Hellier. Video coding using learned latent gan compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2239–2248, 2022. 1
- [28] 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. 7
- [29] K. Tian, Y. Guan, J. Xiang, J. Zhang, X. Han, and W. Yang. Effortless cross-platform video codec: A codebook-based method. *arXiv preprint arXiv:2310.10292*, 2023. 3
- [30] C. Wu, N. Singhal, and P. Krahenbuhl. Video compression through image interpolation. In *ECCV*, pages 416–431, 2018. 3
- [31] J. Wu, L. Liu, Y. Tan, J. Hao, and X. Zhang. Promptus: Can prompts streaming replace video streaming with stable diffusion. *arXiv preprint arXiv:2405.20032*, 2024. 1, 3, 4, 6, 7, 8
- [32] Y. Xu, T. Park, R. Zhang, Y. Zhou, E. Shechtman, F. Liu, J. Huang, and D. Liu. Videogigagan: Towards detail-rich video super-resolution. *arXiv preprint arXiv:2404.12388*, 2024. 3
- [33] R. Yang and S. Mandt. Lossy image compression with conditional diffusion models. *NeurIPS*, 36, 2024. 2, 3
- [34] R. Yang, F. Mentzer, L. V. Gool, and R. Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. In *CVPR*, pages 6628–6637, 2020. 3
- [35] M. Zhao, R. Wang, F. Bao, C. Li, and J. Zhu. Controlvideo: Conditional control for one-shot text-driven video editing and beyond. *arXiv preprint arXiv:2305.17098*, 2023. 3, 4
- [36] S. Zhou, P. Yang, J. Wang, Y. Luo, and C. C. Loy. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In *CVPR*, pages 2535–2545, 2024. 3