Hybrid Video Diffusion Models with 2D Triplane and 3D Wavelet Representation

Kihong Kim [1], Haneol Lee [2], Jihye Park [3], Seyeon Kim [3],
Kwanghee Lee [1], Seungryong Kim [†3], and Jaejun Yoo [†2]

[1]VIVE STUDIOS   [2]UNIST   [3]Korea University
{hxngiee,lkwanghee}@gmail.com
{haneollee,jaejun.yoo}@unist.ac.kr
{ghp1112,kimse2878,seungryong_kim}@korea.ac.kr
https://hxngiee.github.io/HVDM

Abstract. Generating high-quality videos that synthesize desired realistic content is a challenging task due to their intricate high dimensionality and complexity. Several recent diffusion-based methods have shown comparable performance by compressing videos to a lower-dimensional latent space, using traditional video autoencoder architecture. However, such method that employ standard frame-wise 2D or 3D convolution fail to fully exploit the spatio-temporal nature of videos. To address this issue, we propose a novel hybrid video diffusion model, called **HVDM**, which can capture spatio-temporal dependencies more effectively. HVDM is trained by a hybrid video autoencoder which extracts a disentangled representation of the video including: (i) a global context information captured by a 2D projected latent, (ii) a local volume information captured by 3D convolutions with wavelet decomposition, and (iii) a frequency information for improving the video reconstruction. Based on this disentangled representation, our hybrid autoencoder provide a more comprehensive video latent enriching the generated videos with fine structures and details. Experiments on standard video generation benchmarks such as UCF101, SkyTimelapse, and TaiChi demonstrate that the proposed approach achieves state-of-the-art video generation quality, showing a wide range of video applications (e.g., long video generation, image-to-video, and video dynamics control). The source code and pre-trained models will be publicly available once the paper is accepted.

Keywords: Video Diffusion · Wavelet Transform · Generative Models

1   Introduction

Video generation has received a lot of attention in the computer vision and graphics fields due to its broad applications in making films, creating animations, simulating environments, etc [1,5,6,10,23]. Nontheless, video generation remains a chellenging task because of the high-dimensionality and complexity inherent


[†]Corresponding author

in video data. The primary difficulty lies in producing high-quality video frames that are both visually appealing and dynamically coherent.

By leveraging the recent advancements of image-based diffusion models [14, 15], a variety of video diffusion methods have emerged [17, 27, 31]. These methods show considerable promise in modeling video distributions and synthesizing visually plausible results. However, they often face efficiency challenge due to operating in pixel space. To address this challenge, several recent latent video diffusion models have been developed, training within a compressed latent space to reduce computational demands [2, 11, 13, 56, 56]. While these latent diffusion models exhibit improved scaling properties with reduced complexity, the potential of novel autoencoder designs remains largely unexplored. There exist conventional approaches, employing standard 2D or 3D CNN-based autoencoders for video generation [13, 17, 19, 35]. Yet, these CNN-based architectures struggle with effectively managing the overall spatio-temporal structure of videos. While transformers might be considered as a solution, their direct application to video representation learning can lead to significant computational time and memory inefficiency [21, 49].

Unlike the aforementioned approaches, PVDM [53] offers a novel perspective in video generation. PVDM simplifies video complexity by adopting a triplane representation, where video data is factorized into 2D projected latents across different spatio-temporal directions in a latent space. This method has demonstrated improved generation results over previous works such as StyleGAN-based approaches [36] and other diffusion models [17], particularly in effectively representing intricate video data. Despite its progress, they encounter inherent limitations due to the reduction of higher-dimensional video data into lower-dimensional spaces. More specifically, the 2D projected latents struggle to fully recover the 3D structure of video volume, lacking crucial volume information.

Motivated by these observations, we propose a novel hybrid video diffusion model designed to comprehensively capture the spatio-temporal dependencies of video, called **HVDM**. Rather than reinventing the wheel, we leverage these known characteristics to enhance effectiveness in video generation tasks. Our video autoencoder combines transformer for 3D-to-2D projections and the 3D CNNs to effectively utilize both global context and local volume information in video. We first start from a well-known fact that transformers excel at capturing long-range dependencies, and we employ them to acquire the global context of the video. To compensate the limited inductive bias of transformers, we leverage 3D CNNs to provide local volume details, drawing from the recognized advantage of 3D CNNs in capturing short-range spatio-temporal information. More specifically, our proposed autoencoder builds the stengths of both architectures by incorporating global 2D projected and local 3D volume latent representations. These two types of latent representations mutually enhance each other through cross-attention module that operates along both the temporal and spatial axes. Taking this approach further, we pose the question of whether it is possible to exploit frequency information for boosting the video reconstruction process. We find that employing 3D discrete wavelet transform could potentially resolve the
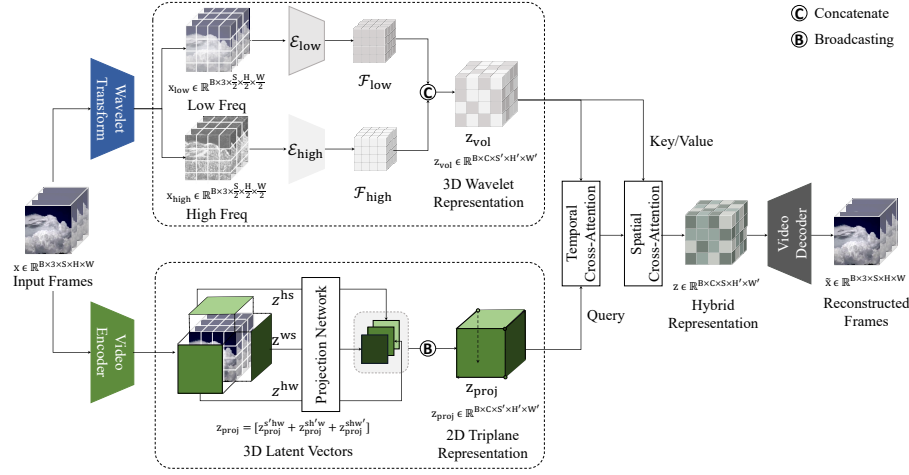
Fig. 1: Overview of our hybrid video autoencoder in HVDM that combines a 2D triplane and 3D volume representation for video encoding. The 2D triplane representation provide global context and 3D volume representation provide local volume information of video. The spatio-temporal cross-attention module incorporates these distinctive features to organizes fine-grained video representation.

problem of limited receptive fields in 3D CNNs. Specifically, considering that the 3D discrete wavelet transform reduces the video's size by half in the frequency domain, it allows us to expand the receptive field without losing information. Additionally, using 3D discrete wavelet transform decompose complex video into different frequency components at multiple levels of details, providing rich video representation. Finally, by training the video autoencoder with both reconstruction and frequency matching loss in the pixel and spectral spaces, frequency matching loss improved visual quality during video reconstruction process. As a result, we observe that the integration of wavelet-based features provides benefit compared to the use of raw video data by enabling more nuanced video encoding.

To validate the effectiveness of our method, we demonstrate a wide range of video applications including unconditional video generation, long video generation, image-to-video, and video dynamics control. Here, we use three benchmark datasets for video generation: UCF-101 [38], SkyTimelapse [51], and TaiChi [34]. Both quantitative and qualitative results demonstrate video generation capabilities of our method, showing superior performance over the state-of-the-art approaches.

2 Related Work

Video generation. Video generation methods have been developed based on deep generative models, such as variational autoencoder (VAEs), generative adversarial networks (GANs), and diffusion models (DMs). Early work [24, 28] attempted to employ VAEs and GANs-based models to generate frames in an

autoregressive manner. However, they fail to produce desired results due to the difficulty of modeling spatial-temporal changes and their results limited to low resolutions. Another work [45] introduce a spatio-temporal convolutional architecture that effectively disentangles the scene's foreground and background. MoCoGAN [40] propose the motion and content decomposed generative adversarial network that generates a video by translating a sequence of random vectors into a sequence of video frames. Although each of these methods show encouraging results for video generation, GAN-based models often suffer from unstable training and fall into mode collapse. In contrast, diffusion models have demonstrated new possibilities for video generation, leveraging a stable training scheme and robust representation.

Diffusion models. Inspired by the success of image-based diffusion models, a series of video diffusion models [2, 11, 25, 30, 50, 56] are proposed. These models leverage the pre-trained image diffusion models and extend their application to videos. While they are capable of efficient learning with just a few computation powers, there are some difficulties in creating consistent and photo-realistic video content. Since they generate videos on a frame-by-frame basis, they face challenges in achieving temporal consistency and producing long video results.

On the other hand, recent videos diffusion models [16, 26, 31, 44] proposed methods for learning video from scratch by adopting various model architectures [9, 29]. While they are capable of generating high fidelity results that are both temporally consistent and realistic, these methods are still limited in computation and memory inefficiency. To overcome these inefficiency, latent diffusion methods [13, 19, 53] have emerged which train the model in low-dimensional latent space while maintaining their perceptual quality.

Triplane representations. The effectiveness of triplane representation has already found in 3D-aware generation works [3, 7, 8, 12, 32, 48]. These triplane representation methods have been shown to offer an effective compression benefits by projecting the 3D volume as 2D-shaped latent features. By taking advantage of these triplane representation, PVDM [53] achieves comparable results in the video generation task, encoding the cubic structure of video pixels. Inspired by this, we present an improved video encoding method that exploit the 2D projected latent with wavelet-based 3D volume latent.

3   Preliminaries

3.1   2D Triplane Representation for Video Diffusion Model

Given a video sample $\mathbf{x} \in \mathbb{R}^{3 \times S \times H \times W}$ with time S, height H, and width W, the triplane autoencoder encodes it as three 2D projected latent $[\mathbf{z}_{\text{proj}}^{\text{hw}}, \mathbf{z}_{\text{proj}}^{\text{ws}}, \mathbf{z}_{\text{proj}}^{\text{hs}}]$ where $\mathbf{z}_{\text{proj}}^{\text{hw}} \in \mathbb{R}^{C \times H' \times W'}$, $\mathbf{z}_{\text{proj}}^{\text{ws}} \in \mathbb{R}^{C \times S \times W'}$, $\mathbf{z}_{\text{proj}}^{\text{hs}} \in \mathbb{R}^{C \times S \times H'}$. We denote C as number of channels, $H' = H/d$, $W' = W/d$ for downsampling factors $d > 1$. Unlike conventional video autoencoder which depends on frame-wise 2D or 3D convolution networks [13, 17, 19, 35], the PVDM [53] effectively decompose

the 3D data using video transformer [4] for projections $[\mathbf{z}_{\text{proj}}^{\text{hw}}, \mathbf{z}_{\text{proj}}^{\text{ws}}, \mathbf{z}_{\text{proj}}^{\text{hs}}]$. The utilization of these latent vectors $[\mathbf{z}_{\text{proj}}^{\text{hw}}, \mathbf{z}_{\text{proj}}^{\text{ws}}, \mathbf{z}_{\text{proj}}^{\text{hs}}]$ offers several advantages in terms of achieving both computation and memory efficiency. Furthermore, their image-like structure enables the efficient video generation from the latent space reducing the dimensionality of the data.

3.2 3D Wavelet Representation

3D Wavelet transform is a classical algorithm widely used in medical imaging and video compression to separate different frequency components within the data [33, 46, 47]. Just as the 2D wavelet transform, given $\mathbf{x} \in \mathbb{R}^{3 \times S \times H \times W}$, the 3D wavelet transform operates along s, h, and w direction decomposing the 3D volume into different subbands. These subbands are factorized into low and high subbands where the low subband express the slowly changing part of the data and the high subband contain fine details capturing sharp edges and rapid transitions within the data.

Specifically, it can be defined as discrete wavelet transform (DWT) and discrete inverse wavelet transform (IWT). In general, Haar wavelets is commonly employed in a variety of real-world applications due to their simplicity. Different types of filters, including low-pass and high-pass filters, divide the volume input $\mathbf{x} \in \mathbb{R}^{3 \times S \times H \times W}$ into distinct frequency sub-bands $\mathbf{x}_{\text{lll}}$, $\mathbf{x}_{\text{llh}}$, $\mathbf{x}_{\text{lhl}}$, $\mathbf{x}_{\text{lhh}}$, $\mathbf{x}_{\text{hll}}$, $\mathbf{x}_{\text{hlh}}$, $\mathbf{x}_{\text{hhl}}$, $\mathbf{x}_{\text{hhh}}$ with a size of $S/2 \times H/2 \times W/2$. Where the low-frequency components provide the long-term averages information, while the high-frequency components provide information about local changes. In Fig. 2, we visualize a one level of 3D discrete wavelet decomposition.
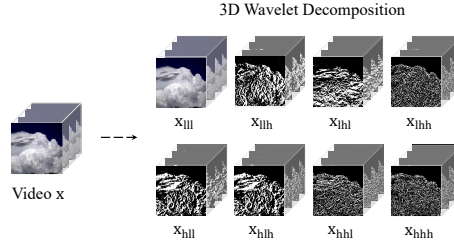
Fig. 2: **Visualization of 3D wavelet transform.** The volume of video is decomposed into eight subband $(\mathbf{x}_{\text{lll}}, \dots, \mathbf{x}_{\text{hhh}})$ including low and high frequency components.

4 Methodology

Our hybrid video autoencoder is illustrated in Fig. 1. We first describe the overall video encoding architecture in Sec 4.1. In the following, we present the training objective which enforce constraints in both pixel space and the wavelet space in Sec 4.2. Then, we formalize the diffusion-based video generation framework in Sec 4.3. Lastly, we introduce various video applications using video diffusion model in Sec 4.4.

4.1 Designing Hybrid Video Autoencoder

To train a video diffusion model generating high quality results, it is necessary to have an expressive video representation that accounts for inclusive information of video data. To satisfy this requirement, we construct hybrid representations by combining 2D triplane and 3D wavelet representation.

2D triplane representation. The 2D projected latent which is originated from PVDM [53] is extracted by 3D-to-2D projections mapping at each spatio-temporal direction to encode 3D video pixels. As shown in Sec 3.1, the 2D image-like latent $[\mathbf{z}_{\text{proj}}^{\text{hw}}, \mathbf{z}_{\text{proj}}^{\text{ws}}, \mathbf{z}_{\text{proj}}^{\text{hs}}]$ can be encoded with a combination of a video encoder [4] and a small transformer [43]. Formally, the projected latent is formulated as follows:

$$
\begin{aligned}
[u_{\text{shw}}] &:= \mathcal{T}_{\text{video}}^{\text{shw}}(\mathbf{x}), \\
\mathbf{z}_{\text{proj}}^{\text{hw}} &:= \mathcal{T}_{\text{proj}}^{\text{s}}(u_{1\text{hw}}, \ldots, u_{\text{Shw}}), \\
\mathbf{z}_{\text{proj}}^{\text{ws}} &:= \mathcal{T}_{\text{proj}}^{\text{h}}(u_{\text{s1w}}, \ldots, u_{\text{sH}'\text{w}}), \\
\mathbf{z}_{\text{proj}}^{\text{hs}} &:= \mathcal{T}_{\text{proj}}^{\text{w}}(u_{\text{sh1}}, \ldots, u_{\text{shW}'}),
\end{aligned}
\tag{1}
$$

where $\mathcal{T}_{\text{video}}^{\text{shw}}$ denotes a video-to-3D-latent encoder such as video transformer [4] and $\mathcal{T}_{\text{proj}} = [\mathcal{T}_{\text{proj}}^{\text{s}}, \mathcal{T}_{\text{proj}}^{\text{h}}, \mathcal{T}_{\text{proj}}^{\text{w}}]$ denotes a small transformer [43] for projections. While projected latent provide a global information of video with a computational efficiency, this transformer architecture projecting a triplane leads to lossy compression due to the lack of local volume information. Moreover, the weak inductive bias of the transformer leads to a slower convergence speed for the model. To overcome these limitations, we transformed the 2D projected latent into a 3D volume and complemented it with the 3D volume representation. Converting a 2D projected latent into a 3D volume $\mathbf{z}_{\text{proj}}$ is formulated as follows:

$$
\begin{aligned}
\mathbf{z}_{\text{proj}}^{\text{s}'\text{hw}} &:= [\mathbf{z}_{\text{proj}}^{1\text{hw}}, \ldots, \mathbf{z}_{\text{proj}}^{\text{Shw}}], \\
\mathbf{z}_{\text{proj}}^{\text{sh}'\text{w}} &:= [\mathbf{z}_{\text{proj}}^{\text{s1w}}, \ldots, \mathbf{z}_{\text{proj}}^{\text{sH}'\text{w}}], \\
\mathbf{z}_{\text{proj}}^{\text{shw}'} &:= [\mathbf{z}_{\text{proj}}^{\text{sh1}}, \ldots, \mathbf{z}_{\text{proj}}^{\text{shW}'}], \\
\mathbf{z}_{\text{proj}} &:= [\mathbf{z}_{\text{proj}}^{\text{s}'\text{hw}} + \mathbf{z}_{\text{proj}}^{\text{sh}'\text{w}} + \mathbf{z}_{\text{proj}}^{\text{shw}'}].
\end{aligned}
\tag{2}
$$

3D wavelet representation. 3D CNN based volume encoder $\mathcal{E}_{\text{low}}, \mathcal{E}_{\text{high}}$ extract local spatio-temporal features in videos. These volumetric local features are exploited with the 2D projected latent $\mathbf{z}_{\text{proj}}$ to reorganize fine-grained features. Instead of utilizing the unprocessed RGB video frame directly in the volume encoder, we employ the decomposed wavelet features as input. These decomposed wavelet features extracted from the 3D wavelet transform $\mathcal{W}$ provide coarse and detailed volume coefficient suited for video encoding. Extracting 3D wavelet latent can be formulated as follows:

$$
\begin{aligned}
\mathcal{F}_{\text{low}} &= \mathcal{E}_{\text{low}}(\mathbf{x}_{\text{lll}}), \\
\mathcal{F}_{\text{high}} &= \mathcal{E}_{\text{high}}(\text{Cat}(\mathbf{x}_{\text{llh}}, \cdots, \mathbf{x}_{\text{hhh}})), \\
\mathbf{z}_{\text{vol}} &= \text{Cat}(\mathcal{F}_{\text{low}}, \mathcal{F}_{\text{high}}).
\end{aligned}
\tag{3}
$$

where $\mathbf{x}_{\text{lll}}$ denotes decomposed low-pass subband and the others $\mathbf{x}_{\text{llh}}, \ldots, X_{\text{hhh}}$ denotes high-pass subband. The low-pass subband and high-pass subband are encoded to provide the coarse content and fine motion information.

Fusing 2D triplane and 3D wavelet representations. As shown in Fig. 1, these 2D projected and 3D volume latent are integrated by cross-attention operation across the temporal and spatial axis. In this way, we can reinforce the volume-related inductive bias and rearrange fine-grained video latent for video generation. Formally, this cross-attention process can be formulated as follows:

$$
\begin{aligned}
\mathbf{z}^i &= \text{CA}(\mathbf{z}^i_{\text{proj}}, \mathbf{z}_{\text{vol}}) + \mathbf{z}^i_{\text{proj}}, \\
\mathbf{z}^i &= \text{FFN}(\mathbf{z}^i) + \mathbf{z}^i, \\
\tilde{\mathbf{z}}^i &= \text{CA}(\mathbf{z}^i, \mathbf{z}_{\text{vol}}) + \mathbf{z}^i, \\
\mathbf{z}^{i+1} &= \text{FFN}(\tilde{\mathbf{z}}^i) + \tilde{\mathbf{z}}^i,
\end{aligned}
\tag{4}
$$

where $\text{CA}(a, b)$ refers to Cross-Attention, with a representing the query and b representing the key and value. Additionally, FFN stands for FeedForward Network. Note that the input $\mathbf{z}^0_{\text{proj}} = \mathbf{z}_{\text{proj}}$ and the final output $\mathbf{z}$. Then, we reconstruct the video $\tilde{\mathbf{x}}$ with video decoder $\mathcal{D}$ based on the refined hybrid features.

$$
\tilde{\mathbf{x}} = \mathcal{D}(\mathbf{z}),
\tag{5}
$$

where $\mathbf{z} \in \mathbb{R}^{C \times S \times H' \times W'}$.

4.2   Loss function

Our hybrid video autoencoder consists of three loss functions. In the following, we provide a brief overview and explain the training loss functions. Our training loss functions enable the preservation of frequency information during the video reconstruction process. Specifically, we enforce constraints both in the pixel space and the wavelet space.

Reconstruction loss. The reconstruction loss $\mathcal{L}_{rec}$ denotes the similarity between the original input video and output video reconstructed by a model. The reconstruction loss term enforce the model to learn meaningful features in the encoding phase so that it can accurately reconstruct the input data in the decoding phase. We employ mean absolute error for reconstruction:

$$
\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}} \left[\| \mathbf{x} - \tilde{\mathbf{x}} \|_1 \right].
\tag{6}
$$

Perceptual loss. The perceptual loss measures the difference between the feature map of input and a generated output in terms of perceptual similarity. Typically, this loss entails the extraction of feature maps from a pre-trained neural network. In contrast to reconstruction loss, which focuses solely on the pixel-level differences between videos, perceptual loss takes into account high level features such as textures and structures that humans perceive.

$$
\mathcal{L}_{\text{lpips}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}} \left[\| f(\mathbf{x}) - f(\tilde{\mathbf{x}}) \|_1 \right].
\tag{7}
$$

where f denotes the perceptual feature extractor.

Frequency matching loss. Conventional methods typically rely on pixel-level losses in the spatial-temporal space. Since these loss functions closely match the original videos in pixel domain, the generated video's quality may not be sufficient potentially failing to capture the frequency information. Our frequency matching loss is designed to facilitate the model can learn frequency features with less optimization difficulty. This loss not only helps the model to recover missing frequency components but also enhances the overall reconstruction quality. We define the frequency matching loss as a reconstruction term in the wavelet domain:

$$\mathcal{L}_{\text{freq}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}} \left[\|\mathcal{W}_{\mathbf{d}}(\mathbf{x}) - \mathcal{W}_{\mathbf{d}}(\tilde{\mathbf{x}})\|_1 \right], \mathbf{d} \in \{\text{s}, \text{h}, \text{w}\}. \tag{8}$$

where $\mathcal{W}_{\mathbf{d}}$ denotes three-dimensional discrete wavelet transform and $\mathbf{d}$ denotes the dimensions from which the wavelet transform is performed.

Overall loss. Taking into account all the losses, the overall loss is formulated as follows:

$$\mathcal{L} = \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{lpips}} \mathcal{L}_{\text{lpips}} + \lambda_{\text{freq}} \mathcal{L}_{\text{freq}}, \tag{9}$$

where $\lambda_{\text{rec}} = \lambda_{\text{lpips}} = 1$ and $\lambda_{\text{freq}} = 0.5$ in our experiments.

Note that, unlike previous works [5,53] that focus on improving the temporal dynamics of generated videos through the use of a 3D discriminator, our video autoencoder achieves temporal realism in the reconstructed videos without the need for adversarial training.

4.3   Diffusion Model-based Video Generator

Leveraging the hybrid latent extracted from the video autoencoder, we train a diffusion-based video generator which learn the latent distribution $p(\mathbf{z})$ through the reverse process of the Markov Chain.

Our diffusion model consists of two processes: the forward process and the reverse process. In forward process, the model gradually transforms the latent $\mathbf{z}_0 \sim p(\mathbf{z}_0)$ toward a Gaussian distribution as formulated below:
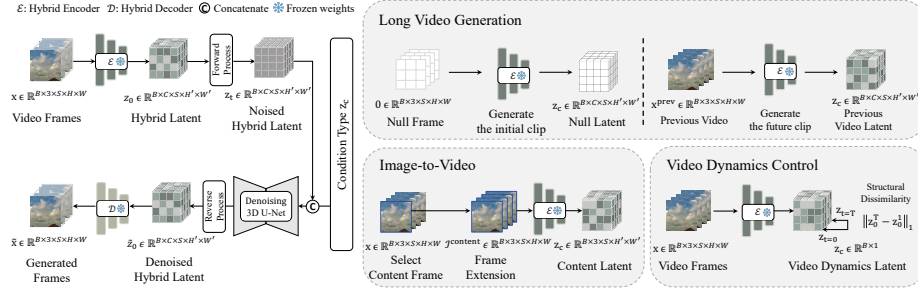
$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I}),$$
$$q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1-\bar{\alpha}_t)\mathbf{I}). \tag{10}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$ and β represents a pre-defined schedule for the variance.

In the reverse process, the model is trained to gradually denoise the latent via the parameterized Gaussian transition.

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, t), \sigma_t^2\mathbf{I}), \tag{11}$$

$$\mu_\theta(\mathbf{z}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{z}_t, t)\right), \tag{12}$$

Fig. 3: Applications for diverse video generator by our proposed HVDM. Our HVDM is adaptable for diverse video generation tasks depending on the type of condition latent z_c. During training process, the condition latent z_c is extracted from the video and jointly trained with the noised hybrid video latent z_t. During the sampling process, these condition latent are supported to enable various video generation tasks, such as long video generation, image-to-video, and video dynamics control.

where $\epsilon_\theta(\cdot)$ is a trainable denoising diffusion model which recover a noisy sample $\mathbf{z}_t$. The denoising diffusion model is trained to predict the noise with the following simplified objective:

$$\mathcal{L}_{\text{simple}}(\theta) := \left\| \epsilon_\theta(\mathbf{z}_t, t) - \epsilon \right\|_2^2, \tag{13}$$

where $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$.

To generate video samples from the latent space, we train a denoising diffusion model which follow the standard 3D U-Net architecture. The 3D U-Net architecture is composed of 3D convolution layers combined with skip connections. We employ a simplified 3D U-Net to denoise the latent $\mathbf{z}$ for the purpose of verifying the effectiveness of our video autoencoder.

4.4   Applications for Diverse Video Generator

We demonstrate that our method can be extended to a variety of video applications such as long video generation, image-to-video, and video dynamics control.

Long video generation. To facilitate long video generation, our work follows the traditional frame prediction approach [13, 44, 53] where previous frame conditions are combined with future frames. Given two consecutive video clips $[\mathbf{x}^{\text{prev}}, \mathbf{x}^{\text{next}}]$ of $\mathbf{x}^{\text{prev}}, \mathbf{x}^{\text{next}} \in \mathbb{R}^{3 \times S \times H \times W}$, we can train the model to generate initial clips and future clips in a unified manner. More specifically, we can generate an initial clip with a null latent and extend the video by composing the condition with previous frame latent as follow:

$$\mathbf{z}_c^{\text{long}} = \begin{cases} \mathcal{E}(\mathbf{x}^{\text{prev}}), & \text{if future clip,} \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $\mathcal{E}$ denotes our pre-trained hybrid encoder.

Image-to-video. Image-to-video generation aims to synthesize a coherent video starting from an image. In general, videos are composed of a series of consecutive images, where the first frame serves as a representative depiction of the overall content within the video. To facilitate the process of image-to-video generation without introducing an additional image encoder, we repeatedly stack the first frame of the video on the temporal axis. Then, we use the latent $\mathcal{I}^{\text{content}}$ extracted from our video encoder as the content condition. By leveraging the latent $\mathcal{I}^{\text{content}}$ which capture the common content, we can effectively guide the model to generate coherent video sequence as follow:

$$\begin{aligned} \mathcal{I}^{\text{content}} &= [\mathbf{x}_0^1, \dots, \mathbf{x}_0^1] \in \mathbb{R}^{3 \times S \times H \times W}, \\ \mathbf{z}_c^{\text{img2vid}} &= \mathcal{E}(\mathcal{I}^{\text{content}}). \end{aligned} \tag{15}$$

Video dynamics control. Empowering users to control motion intensity in video creation provides flexibility with an improved user experience. Unfortunately, it was difficult to obtain motion-related labels from existing video collections for training diffusion models. To enable the alteration of motion intensity, we devise a method that learns motion dynamics with inter-frame variance. Specifically, we define the level of motion alteration as the structural dissimilarity between the semantic latent of the start and end frames. Since the structural dissimilarity encompasses both semantic and motion variance, we employ it as guiding cues to control motion without explicit motion labeled video. Formally, our structural dissimilarity is defined as follow:

$$\mathbf{z}_c^{\text{dynamics}} = \|\mathbf{z}_0^{\text{T}} - \mathbf{z}_0^1\|_1. \tag{16}$$

Inspired by the classifier-free guidance techniques, we train a single diffusion model to jointly learn both the unconditional distribution $p(\mathbf{x})$ and the conditional distribution $p(\mathbf{x}^2|\mathbf{x}^1)$. In particular, we train the conditional distribution $p(\mathbf{x}^2|\mathbf{x}^1)$ with condition-specific information (e.g., previous frame or content or dynamics latent) and the unconditional distribution using a null condition (e.g., $\mathbf{z}_c = 0$). The optimized objective can be formulated as follow:

$$\mathbb{E}_{\mathbf{x}_0, \epsilon, t}\left[\lambda||\epsilon - \epsilon_\theta(\mathbf{z}_t, \mathbf{z}_c, t)||_2^2 + (1 - \lambda)||\epsilon - \epsilon_\theta(\mathbf{z}_t, 0, t)||_2^2\right], \tag{17}$$

where $\mathbf{z}_0 = \mathcal{E}(\mathbf{x}_0), \mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$ and $\lambda \in (0, 1)$ is a hyperparameter that guide a model to learn an unconditional and conditional distributions.

5 Experiments

5.1 Training Details

Datasets. We conduct the experiments on three commonly-used benchmark datasets for the video generation task: UCF-101 [38], SkyTimelapse [51], and TaiChi [34]. UCF-101 [38] contains 13,320 videos from 101 action classes, offering a diverse range of human actions. SkyTimelapse [51] is composed of 5,000

Method	UCF-101				SkyTimelapse				TaiChi			
	R-FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑	R-FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑	R-FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
LVDM [13]	-	-	-	-	74.92	0.091	31.21	0.778	102.68	0.112	27.63	0.729
PVDM [53]	27.03	0.095	27.03	0.742	35.96	0.061	32.71	0.826	61.61	0.068	26.87	0.715
HVDM (ours)	**5.35**	**0.038**	**34.00**	**0.915**	**20.76**	**0.046**	**35.04**	**0.891**	**21.51**	**0.025**	**33.51**	**0.916**

Table 1: Quantitative comparison of reconstruction results on UCF-101 [38], SkyTimelapse [51], and TaiChi datasets [34]. Our video autoencoder shows significantly superior performance across various metrics, providing more fine-grained comparision.

Fig. 4: Qualitative reconstruction results on SkyTimelapse [51] dataset [34]. Our HVDM produces distinct and sharp edge details and faithfully expresses a fine structures in a video sequence. Additional samples are included in appendix.

videos containing dynamic sky scenes. TaiChi [34] consists of full human bodies performing Tai-Chi actions.

Training. The video autoencoder component for 2D projected latent stem from the PVDM [53], but the other wavelet-based frequency component is added. We train our models on 4 NVIDIA A100 GPUs for each autoencoding and diffusion process. See Appendix A for the details.

5.2   Evaluation Details

Baseline models. We compare our HVDM with the following video generation models: VideoGPT [52], MoCoGAN [40], MoCoGAN-HD [39], DIGAN [54], StyleGAN-V [36], LVDM [13], and PVDM [53]. We utilize official model weights and conduct training using the official code when the model weights are not available.

Quantitative evaluations. To evaluate reconstruction quality, we use R-FVD [42], LPIPS [55], PSNR [20], and SSIM [18]. Here, R-FVD denotes FVD between reconstructed videos and the ground-truth videos. As for the evaluation of generation quality, we measure FVD [42] and KVD [41]. Considering different clip

	VideoGPT	MoCoGAN	+ StyleGAN2	MoCoGAN-HD	DIGAN	StyleGAN-V	LVDM†	PVDM†	HVDM (Ours)
				Train Split					
UCF101 - FVD_{16}	2880.6	2886.8	1821.4	1729.6	1630.2	1431.0	N/A	399.4	**303.1**
UCF101 - FVD_{128}	N/A	3679.0	2311.3	2606.5	2293.7	1773.4	N/A	**505.0**	549.7
SkyTimelapse - FVD_{16}	222.7	206.6	85.88	164.1	83.11	79.52	318.9	63.5	**42.8**
SkyTimelapse - FVD_{128}	N/A	575.9	272.8	878.1	196.7	197.0	N/A	134.7	**124.9**

	LVDM†	PVDM†	HVDM (Ours)		LVDM†	PVDM*	HVDM (Ours)
		Train Split				Train+Test Split	
UCF101 - KVD_{16}	N/A	35.5	**23.6**	TaiChi - FVD_{16}	120.9	267.0	**77.0**
UCF101 - KVD_{128}	N/A	37.7	**34.4**	TaiChi - FVD_{128}	N/A	339.2	**258.5**
SkyTimelapse - KVD_{16}	21.73	3.1	**1.5**	TaiChi - KVD_{16}	14.7	66.5	**11.9**
SkyTimelapse - KVD_{128}	N/A	6.0	**3.5**	TaiChi - KVD_{128}	N/A	120.1	**77.8**

Table 2: Quantitative comparison of video generation models on UCF-101 [38], SkyTimelapse [51], and TaiChi datasets [34]. We evaluate FVD and KVD metrics (the lower values are better) and the best score are marked as bold. The overall score of other baselines are referenced from StyleGAN-V [37]. Methods marked with † denote the score is computed using official pretrained weights and * denote the score is reproduced using official codes. We report evaluation details and sampling settings on Appendix B.

Fig. 5: Generated video sample by our HVDM on UCF-101 [38], SkyTimelapse [51], and TaiChi datasets [34].

lengths, we denote FVD_{16} and KVD_{16} scores, as well as FVD_{128} and KVD_{128} scores for video clips with a length of 16 and 128. For a fair comparison, we follow the evaluation protocol of StyleGAN-V [37]; i.e., we use 2,048 real/fake video clips for evaluating FVD. See Appendix B for the details.

5.3   Comparison Results

Reconstruction results. Fig. 4 and Appendix F.1 show the performance of the reconstructed videos from our autoencoder and other diffusion-based video generative models. Our autoencoder presents a more accurate and clear reconstructed video compared to other baseline methods. In the Tab. 1, we demonstrate that our autoencoder achieves superior performance in all metrics for the three datasets.

Generation results. We perform video generation on three datasets and compare the results with the other video generation methods. As shown in Fig. 5, our method is capable of generating not only monotonic motion videos such as SkyTimelapse [51] and TaiChi [34] but also dynamic motion video UCF-101 [38],

Backbone	Proj.	Adv.	**z** shape dimension	R-FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
Video Transformer [4]	✓	-	$4 \times (32 \times 32 + 16 \times 32 + 16 \times 32)$	180.21	0.092	28.12	0.782
Video Transformer [4]	✓	✓	$4 \times (32 \times 32 + 16 \times 32 + 16 \times 32)$	27.03	0.095	27.03	0.742
3D CNN [22]	-	-	$4 \times 16 \times 32 \times 32$	86.73	0.057	31.76	0.852
3D CNN [22]	-	✓	$4 \times 16 \times 32 \times 32$	21.69	0.064	30.39	0.837
Ours	**✓**	**-**	$\mathbf{4 \times 16 \times 32 \times 32}$	**5.35**	**0.038**	**34.00**	**0.915**

Table 3: Ablation study on various backbone architectures

which contain complex action categories. Note that synthesizing high quality videos on a complex dataset like UCF-101 [38] has been a challenging problem where existing video generation method often struggle with producing natural motions with good quality. As shown in Tab. 2, our method achieves superior performance in the benchmark datasets. See Appendix F for more samples.

5.4   Interpreting the frequency supervision
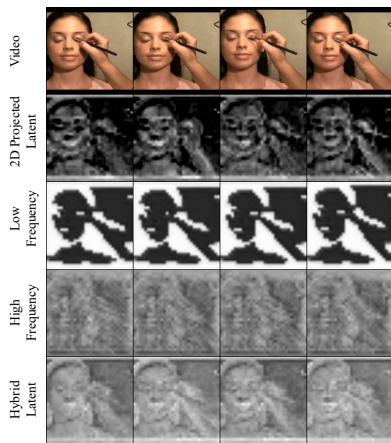
Fig. 6: Visualization of our hybrid representations.

We perform visualization to understand how hybrid representation can help autoencoder to encode the 3D video pixels in Fig. 6. The decomposed 3D wavelet features is divided into low frequency and high frequency components, where each component conduct cross-attention operation with the 2D projected latent. As shown in Fig. 6, low frequency component captures static and dynamic regions and high frequency component capture the sharp edges and important details of video. To refine the ambiguity of the 2D project latent which contain global and abstract information, we incorporate these latent with frequency components. As a result, our autoencoder figures out where the objects are located and how object's motion occurs in the video.

5.5   Ablation study on backbone architectures

We conduct verification of performance based on the autoencoder's backbone architecture on UCF-101 dataset. As shown in Tab. 3, our hybrid autoencoder demonstrates effectiveness in capturing both perceptual details and temporal coherence in the video. We can observe that 3D CNN-based autoencoder show better performance compared to the 2D triplane-based autoencoder since it has more latent dimensions. Although the 2D triplane-based autoencoder effectively compresses latent dimensions, enhancing temporal coherence necessitates adversarial training with a 3D discriminator. Our hybrid autoencoder integrates the strengths of both architectures, achieving superior performance.

5.6   Ablation study on fusing modules

We verify several experiments to assess how the fusing module affects the performance of our hybrid autoencoder on UCF-101 dataset. For feature fusion, we adopted two different modules: 1) the concatenation module 2) the cross-attention module. We also validated the effectiveness of wavelet decomposition by comparing the results which use raw RGB video as input. As shown in Tab. 4, the experimental results show that the cross-attention module and wavelet-decomposed components contribute to the improvement of reconstruction performance. We analyze that utilizing the temporal and spatial attention module facilitate the integration of different 2D triplane and 3D wavelet representation through the comparison of features at different spatial locations.

Fusion module	R-FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑	Fusion module	R-FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
Baseline	27.03	0.095	27.03	0.742	Baseline + RGB (Cross-attn.)	6.32	0.038	33.79	0.908
Baseline + RGB (Concat.)	20.37	0.063	31.02	0.831	Baseline + Low Freq. (Cross-attn.)	6.87	0.040	33.69	0.909
Baseline + Low Freq. (Concat.)	34.39	0.078	30.50	0.816	Baseline + High Freq. (Cross-attn.)	6.97	0.047	33.20	0.899
Baseline + High Freq. (Concat.)	14.96	0.061	31.32	0.838	**Ours**	**5.35**	**0.037**	**34.00**	**0.915**
Baseline + Low & High Freq. (Concat.)	9.92	0.051	32.69	0.871					

Table 4: Ablation study on feature fusing module.

5.7   Ablation study on frequency matching loss

We validate the effectiveness of frequency matching loss (FML) on UCF-101 dataset in Tab. 5. The result shows that utilizing the FML term leads to a remarkable reduction in the FVD score by approximately 2.73 points demonstrating its effectiveness in enhancing the temporal coherency. Furthermore, evaluation metrics related to perceptual details also improved to 0.003 points for LPIPS, 0.33 points for PSNR, and 0.023 points for SSIM respectively. Based on these results, the FML verifies that aligning frequency components allow the HVDM to enhance temporal changes and perceptual quality of generated videos.

Method	R-FVD ↓	LPIPS ↓	PSNR ↑	SSIM ↑
HVDM (w/o FML)	8.08	0.041	33.67	0.892
HVDM (w/ FML)	**5.35**	**0.038**	**34.00**	**0.915**

Table 5: Ablation study on frequency matching loss.

6   Conclusion

We proposed a novel hybrid video autoencoder for video generation using diffusion model, dubbed HVDM. We investigated the effectiveness of our hybrid architecture that combines 2D projected and 3D volume representation, leveraging the strengths of frequency matching loss. By incorporating these representations with spatio-temporal cross-attention, HVDM generates high-quality videos with improved realism. Our extensive experiments on benchmark datasets demonstrated the superior performance, compared to the other baseline methods.

Acknowledgements

References

1. Ajay, A., Han, S., Du, Y., Li, S., Gupta, A., Jaakkola, T., Tenenbaum, J., Kaelbling, L., Srivastava, A., Agrawal, P.: Compositional foundation models for hierarchical planning. arXiv preprint arXiv:2309.08587 (2023)

2. An, J., Zhang, S., Yang, H., Gupta, S., Huang, J.B., Luo, J., Yin, X.: Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. arXiv preprint arXiv:2304.08477 (2023)

3. Anciukevičius, T., Xu, Z., Fisher, M., Henderson, P., Bilen, H., Mitra, N.J., Guerrero, P.: Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12608–12618 (2023)

4. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML (2021)

5. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22563–22575 (2023)

6. Ceylan, D., Huang, C.H.P., Mitra, N.J.: Pix2video: Video editing using image diffusion. arXiv preprint arXiv:2303.12688 (2023)

7. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16123–16133 (2022)

8. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: Pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In: CVPR (2021)

9. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19. pp. 424–432. Springer (2016)

10. Dai, Y., Yang, M., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D., Abbeel, P.: Learning universal policies via text-guided video generation. arXiv preprint arXiv:2302.00111 (2023)
11. Esser, P., Chiu, J., Atighehchian, P., Granskog, J., Germanidis, A.: Structure and content-guided video synthesis with diffusion models. arXiv preprint arXiv:2302.03011 (2023)
12. Gu, J., Trevithick, A., Lin, K.E., Susskind, J.M., Theobalt, C., Liu, L., Ramamoorthi, R.: Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In: International Conference on Machine Learning. pp. 11808–11826. PMLR (2023)
13. He, Y., Yang, T., Zhang, Y., Shan, Y., Chen, Q.: Latent video diffusion models for high-fidelity video generation with arbitrary lengths. arXiv preprint arXiv:2211.13221 (2022)
14. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., Salimans, T.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (2020)
16. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models, 2022. URL https://arxiv. org/abs/2204.03458
17. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. arXiv preprint arXiv:2204.03458 (2022)
18. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th international conference on pattern recognition. pp. 2366–2369. IEEE (2010)
19. Hu, Y., Chen, Z., Luo, C.: Lamd: Latent motion diffusion for video generation. arXiv preprint arXiv:2304.11603 (2023)
20. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electronics letters **44**(13), 800–801 (2008)
21. Kalchbrenner, N., Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., Kavukcuoglu, K.: Video pixel networks. In: ICML (2017)
22. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
23. Khachatryan, L., Movsisyan, A., Tadevosyan, V., Henschel, R., Wang, Z., Navasardyan, S., Shi, H.: Text2video-zero: Text-to-image diffusion models are zero-shot video generators. arXiv preprint arXiv:2303.13439 (2023)
24. Li, Y., Min, M., Shen, D., Carlson, D., Carin, L.: Video generation from text. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
25. Liu, S., Zhang, Y., Li, W., Lin, Z., Jia, J.: Video-p2p: Video editing with cross-attention control. arXiv preprint arXiv:2303.04761 (2023)
26. Ma, X., Wang, Y., Jia, G., Chen, X., Liu, Z., Li, Y.F., Chen, C., Qiao, Y.: Latte: Latent diffusion transformer for video generation. arXiv preprint arXiv:2401.03048 (2024)
27. Mei, K., Patel, V.M.: Vidm: Video implicit diffusion models. In: AAAI (2023)
28. Pan, Y., Qiu, Z., Yao, T., Li, H., Mei, T.: To create what you tell: Generating videos from captions. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1789–1798 (2017)
29. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023)

30. Qi, C., Cun, X., Zhang, Y., Lei, C., Wang, X., Shan, Y., Chen, Q.: Fatezero: Fusing attentions for zero-shot text-based video editing. arXiv preprint arXiv:2303.09535 (2023)
31. Ruan, L., Ma, Y., Yang, H., He, H., Liu, B., Fu, J., Yuan, N.J., Jin, Q., Guo, B.: Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10219–10228 (2023)
32. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
33. Secker, A., Taubman, D.: Highly scalable video compression using a lifting-based 3d wavelet transform with deformable mesh motion compensation. In: Proceedings. International Conference on Image Processing. vol. 3, pp. 749–752. IEEE (2002)
34. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. In: NeurIPS (2019)
35. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. In: ICLR (2023)
36. Skorokhodov, I., Tulyakov, S., Elhoseiny, M.: StyleGAN-V: A continuous video generator with the price, image quality and perks of StyleGAN2. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
37. Skorokhodov, I., Tulyakov, S., Elhoseiny, M.: Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3626–3636 (2022)
38. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
39. Tian, Y., Ren, J., Chai, M., Olszewski, K., Peng, X., Metaxas, D.N., Tulyakov, S.: A good image generator is what you need for high-resolution video synthesis. In: ICLR (2021)
40. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: MoCoGAN: Decomposing motion and content for video generation. In: CVPR (2018)
41. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018)
42. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Fvd: A new metric for video generation. In: DGS@ICLR (2019), `https://api.semanticscholar.org/CorpusID:198489709`
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
44. Voleti, V., Jolicoeur-Martineau, A., Pal, C.: Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. Advances in Neural Information Processing Systems 35, 23371–23385 (2022)
45. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: NeurIPS (2016)
46. Wang, B., Wang, Y., Selesnick, I., Vetro, A.: Video coding using 3d dual-tree wavelet transform. EURASIP Journal on Image and Video Processing **2007**, 1–15 (2007)
47. Wang, J., Huang, K.: Medical image compression by using three-dimensional wavelet transformation. IEEE Transactions on Medical Imaging **15**(4), 547–554 (1996)

48. Wang, T., Zhang, B., Zhang, T., Gu, S., Bao, J., Baltrusaitis, T., Shen, J., Chen, D., Wen, F., Chen, Q., et al.: Rodin: A generative model for sculpting 3d digital avatars using diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4563–4573 (2023)
49. Weissenborn, D., Täckström, O., Uszkoreit, J.: Scaling autoregressive video models. arXiv preprint arXiv:1906.02634 (2019)
50. Wu, J.Z., Ge, Y., Wang, X., Lei, W., Gu, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. arXiv preprint arXiv:2212.11565 (2022)
51. Xiong, W., Luo, W., Ma, L., Liu, W., Luo, J.: Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In: CVPR (2018)
52. Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: VideoGPT: Video generation using VQ-VAE and transformers. arXiv preprint arXiv:2104.10157 (2021)
53. Yu, S., Sohn, K., Kim, S., Shin, J.: Video probabilistic diffusion models in projected latent space. arXiv preprint arXiv:2302.07685 (2023)
54. Yu, S., Tack, J., Mo, S., Kim, H., Kim, J., Ha, J.W., Shin, J.: Generating videos with dynamics-aware implicit generative adversarial networks. In: The Tenth International Conference on Learning Representations (2022)
55. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
56. Zhou, D., Wang, W., Yan, H., Lv, W., Zhu, Y., Feng, J.: Magicvideo: Efficient video generation with latent diffusion models. arXiv preprint arXiv:2211.11018 (2022)