# VideoVAE+: Large Motion Video Autoencoding with Cross-modal Video VAE

Yazhou Xing* Yang Fei* Yingqing He*† Jingye Chen Jiaxin Xie
Xiaowei Chi Qifeng Chen†
The Hong Kong University of Science and Technology

## Abstract

*Learning a robust video Variational Autoencoder (VAE) is essential for reducing video redundancy and facilitating efficient video generation. Directly applying image VAEs to individual frames in isolation results in temporal inconsistencies and fails to compress temporal redundancy effectively. Existing works on Video VAEs compress temporal redundancy but struggle to handle videos with large motion effectively. They suffer from issues such as severe image blur and loss of detail in scenarios with large motion. In this paper, we present a powerful video VAE named VideoVAE+ that effectively reconstructs videos with large motion. First, we investigate two architecture choices and propose our simple yet effective architecture with better spatiotemporal joint modeling performance. Second, we propose to leverage the textual information in existing text-to-video datasets and incorporate text guidance during training. The textural guidance is optional during inference. We find that this design enhances the reconstruction quality and preservation of detail. Finally, our models achieve strong performance compared with various baseline approaches in both general videos and large motion videos, demonstrating its effectiveness on the challenging large motion scenarios.*

## 1. Introduction

Given the significant attention in the field of video generation, Latent Video Diffusion Models (LVDMs) [4, 5, 14, 37] have emerged as a popular framework. They have been successfully applied to powerful text-to-video models such as VideoCrafter [7, 8], Sora [6], CogVideoX [31] and Hunyuan Video [27]. Different from directly generating video pixels, LVDMs generate latent representations of videos in a compact latent space. This is achieved by first training a Video VAE to encode videos into this latent space. Thus, Video VAE, as a key and fundamental component of LVDMs, has attracted great attention recently. An effective Video VAE can help to reduce the training costs of video diffusion models while improving the final quality of the generated videos. Initially, a series of studies adopted the image VAE from Stable Diffusion [21] for video generation tasks, such as AnimateDiff [13], MagicVideo [37], VideoCrafter1 [7], VideoCrafter2 [8], and Align-Your-Latents [5]. However, directly adopting an image VAE and compressing video on a frame-by-frame basis leads to temporal flickering due to the lack of temporal correlation. Additionally, the information redundancy along the temporal dimension is not reduced, leading to low training efficiency for subsequent latent video diffusion models. From the introduction of Sora, which compresses videos both temporally and spatially through a Video VAE, a series of studies have emerged that aim to replicate Sora and train their own Video VAEs, including Open Sora [36], Open Sora Plan [17], CV-VAE [35], CogVideo X [31], Easy Animate [30], and Cosmos Tokenizer [20]. Despite their progress in temporal compression, current video VAEs still struggle to handle videos with large motion. They suffer from many problems in large motion scenarios, including motion ghost, low-level temporal flickering, blurring (faces, hands, edges, texts), and motion stuttering (lack of correct temporal transition).

In this work, we investigate the architecture of Video VAE and propose simple yet effective Video VAE models that have robust reconstruction performance on both general videos and videos with large motion. First, we examine two designs for spatial and temporal compression, including simultaneous spatial-temporal (ST) compression and sequential ST compression. We observed that simultaneous ST compression achieves better low-level temporal smoothness and texture stability, while sequential ST compression achieves better motion recovery, particularly in scenarios of large motion. Thus, we propose our architecture that integrates the advantages of both methods and enables effective video detail and motion reconstruction.

Second, we integrate extra information from another modality, *e.g.,* text, into the Video VAE models, and propose the first text-guided Video VAE. The motivation is as follows: (1) We observed that the normally used datasets for text-to-video generation contain text-video pairs, thus

*equal contribution
†corresponding authors

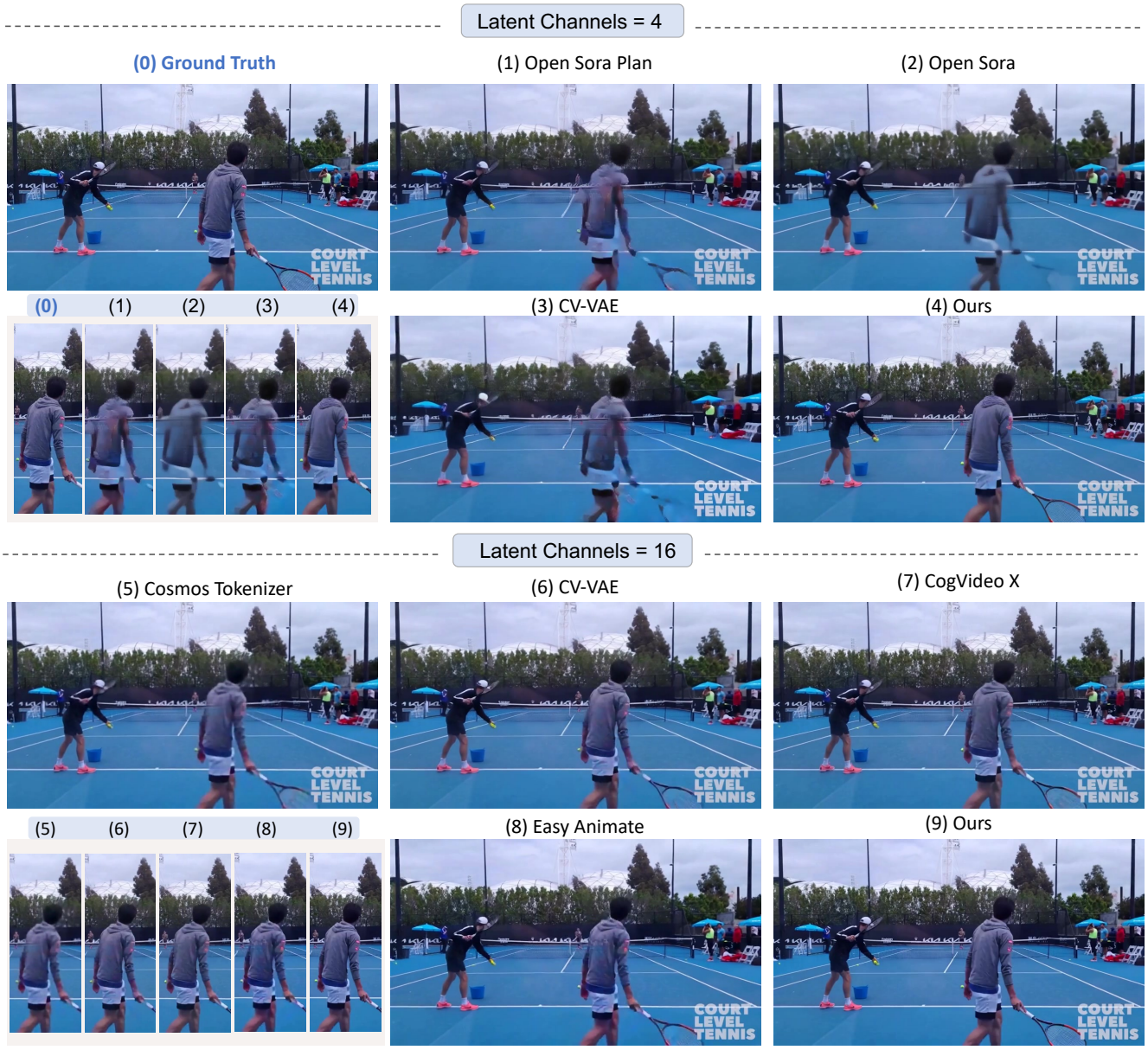


Figure 1. Our reconstruction results compared with a line of three recent strong baseline approaches. The ground truth frame is (0). Our model significantly outperforms previous methods, especially under large motion scenarios such as people doing sports.

the text descriptions for training VideoVAE models are already existing and easy to obtain. (2) Since video compression is inherently lossy, it is challenging to recover all details from the video latent representations without additional supplementary information. As a result, achieving high-quality reconstruction becomes increasingly difficult at higher compression ratios, as finer video details are more likely to be lost. To this end, we conduct experiment on this design, and we discover that leveraging information from other modalities, *e.g.,* text, enhances video reconstruction performance, particularly improving detail reconstruction and boosting performance under higher compression ratio settings. It also helps the model converge during training.

To evaluate Video VAE models in scenarios with large motion, we construct an additional test set containing videos with large motion. Through extensive experiments and comparisons, we demonstrate the superior performance of our models under various settings.

In summary, our contributions are as follows:

- We propose an effective and robust Video VAE, conduct extensive experiments, and achieve state-of-the-art.
- We propose a simple yet effective spatiotemporal modeling architecture for Video VAE.
- We propose the first text-guided video VAE that leverages

the information from other modalities, i.e., text guidance, and shows the improvement of textual guidance.

## 2. Related Work

**Video Variational Autoencoder** Video Variational Autoencoders (VAEs) [16] can be broadly categorized into discrete and continuous types. Discrete video VAEs compress videos into discrete tokens by learning a codebook for quantization and have achieved state-of-the-art performance in video reconstruction, as demonstrated by models like MAGVIT-v2 [33]. However, these VAEs are not suitable for Latent Video Diffusion Models (LVDMs) [15] due to the lack of necessary gradients for backpropagation, which hinders smooth optimization.

In contrast, continuous Video VAEs compress videos into continuous latent representations that are widely adopted in LVDMs. In earlier video generation studies, including Stable Video Diffusion [4], the Video VAE was directly adapted from the image VAE used in Stable Diffusion [21], achieving a compression ratio of $1 \times 8 \times 8$ by processing each frame independently. To further reduce the temporal redundancy, more recent studies [17, 30, 31, 35, 36] have trained their VAEs to achieve a more efficient compression ratio of $4 \times 8 \times 8$.

Despite these advancements, all of the aforementioned video VAEs struggle with accurately reconstructing videos with large motions due primarily to their limited ability to handle the temporal dimension effectively. A high-quality Video VAE that can robustly reconstruct videos with significant motion is critical in the LVDM pipeline, as it ensures efficient latent space compression, maintains temporal coherence and reduces computational overhead [25]. Without a robust VAE, large motions in videos can lead to poor latent representations, negatively impacting the quality and overall performance of the LVDMs.

**Latent Video Diffusion Models** Latent Video Diffusion Models (LVDMs), such as Sora [6], OpenSora [36], Open Sora Plan [17], VideoCrafter1 [7], VideoCrafter2 [8], Latte[19], CogVideoX [31], DynamiCrafter [29] and Vidu [3] have recently achieved significant success in video generation tasks. The general pipeline for these LVDMs consists of two primary steps. First, the raw video is compressed into a latent space via a video Variational Autoencoder (VAE), significantly reducing computational complexity. In the second step, a diffusion model operates within this latent space, learning the desired transformations. The performance of LVDMs is critically dependent on video VAEs, as the quality of the generated video is heavily influenced by the latent space representation and the encoding-decoding capabilities of the VAE.

In image generation tasks, Stable Diffusion [21] has excelled, largely due to its efficient VAE that reconstructs diverse image types with high fidelity. However, no existing VAE in video generation achieves comparable quality, particularly due to challenges in compressing the temporal dimension. This limitation hinders the performance of LVDMs, especially in high-motion scenarios.

## 3. Method

### 3.1. Overview

The video autoencoding problem can be defined as follows. Let $\mathbf{X} \in \mathbb{R}^{C \times T \times H \times W}$ represent a video or image tensor, where $C$, $T$, $H$, and $W$ denote the number of channels, frame(s), height, and width, respectively. We want to train an encoder $\mathcal{E}$ that compresses the input tensor $\mathbf{X}$ into a compact latent representation $\mathbf{Z} \in \mathbb{R}^{C' \times T' \times H' \times W'}$. The learned compact latent $\mathbf{Z}$ can be further reconstructed back to RGB space with decoder $\mathcal{D}$:

$$\mathbf{Z} = \mathcal{E}(\mathbf{X}), \hat{\mathbf{X}} = \mathcal{D}(\mathbf{Z}). \tag{1}$$

Our goal is to design and learn such an autoencoder that can reduce the spatial and temporal dimension of video data in latent space and reconstruct the video with highly spatial and temporal fidelity, especially for large-motion scenarios.

We first examine two inherited video VAE designs from the pre-trained Stable Diffusion model. We then combine the best of two designs and propose our spatiotemporal modeling that can reconstruct high-dynamic contents with fine details. We then investigate the text-conditioned video autoencoding and propose an effective text-guided video VAE architecture. Moreover, we propose a joint image and video compression training method, that enables text-aided joint image and video autoencoding. Our method does not rely on causal convolution as adopted by prior works. Finally, we carefully study the effects of different loss functions on the reconstruction performance and present the state-of-the-art video VAE architecture.

### 3.2. Towards Better Spatiotemporal Modeling

Designing a video VAE that is inherited from a pre-trained 2D spatial VAE is a good practice to leverage the spatial compression prior. There are typically two options to inflate a 2D spatial VAE to its 3D video counterpart.

**Option 1: Simultaneous Spatiotemporal Compression** One common way to inherit the weight from pre-trained 2D VAE is to inflate the 2D spatial blocks to 3D temporal blocks and simultaneously do the spatial and temporal compression. We first examine this design. Specifically, we replace the 2D convolution in SD VAE with 3D convolution of kernel size (1,3,3), whose weights are initialized from the 2D convolution. Then we add an additional temporal convolution layer with kernel size (3,3,3) to learn spatiotemporal

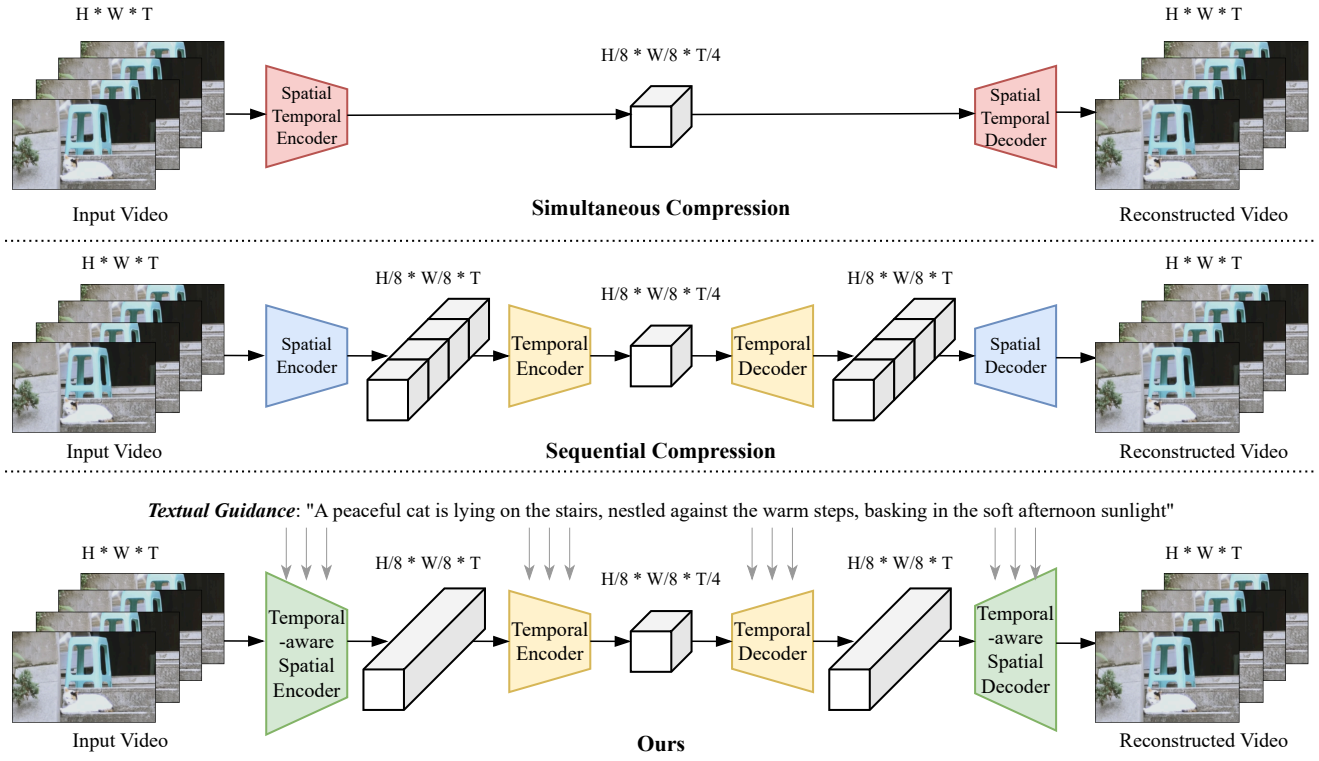


Figure 2. We first investigate two preliminary solutions including simultaneous compression and sequential compression. Simultaneous modeling is achieved by inflating pre-trained 2D spatial VAE to 3D VAE, and simultaneously compress both spatial and temporal information. Sequential modeling indicates first compressing the spatial dimension with a spatial-only 2D encoder and then compressing the temporal information with a temporal encoder. The decoding is symmetrical to the encoding architecture. Our solution is firstly compressing on the spatial dimension, with temporal-aware modelling, and then compressing on the temporal dimension, with a temporal encoder. The decoding is symmetrical to the encoding architecture. Additionally, our VAE also benefits from easy-obtained cross-modality information, i.e., textual guidance, to provide more information to reconstruct fine details.

patterns. In this middle block of the inflated VAE, we inflate the 2D attention to 3D attention and we also include a temporal attention to capture both the spatial and temporal information. We keep other components unchanged to maximumly leverage the learned prior of SD VAE.

**Option 2: Sequential Spatiotemporal Compression** Another reasonable way to cooperate the SD VAE to video VAE is to keep the SD VAE unchanged: first utilize the SD VAE to compress the input video frame-by-frame, and then learn a temporal autoencoding process to further compress the temporal redundancy, as shown in Fig. 2. Specifically, we adopt a lightweight temporal autoencoder for temporal compression. The encoder consists of one convolutional layer to process the input, and two or three 3D ResNet blocks with convolutional downsampling layers to compress the temporal redundancy. Notably, we design the decoder to be asymmetric as the encoder, i.e., there will be two 3D ResNet blocks following each upsampling layer in the decoder. Through this asymmetric design, our decoder can potentially gain some hallucination ability beyond the reconstruction.

Surprisingly, we find this sequential spatiotemporal design can better compress and recover the dynamic of the input video than option 1, but is not good at recovering spatial details, which is proved by consistent improvement under large-motion video autoencoding as shown in Fig. 4.

**Our Solution** We find that simultaneous spatiotemporal compression leads to better detail-recovering capability, and the sequential spatiotemporal compression will exceed at motion-recovering ability. Thus, we propose to combine the best of two worlds, and introduce the two-stage spatiotemporal modeling for video VAE. As the first stage, we inflate the 2D convolution to 3D convolution with kernel size (1,3,3), and similarly to option 1, we add additional temporal convolution layers through 3D convolution. We also include a temporal attention in the middle block of the encoder and decoder, after the spatial cross attention layers. We denote our first-stage model as a temporal-aware spatial autoencoder. Different from option 1, we only compress the spatial information and do not compress the temporal

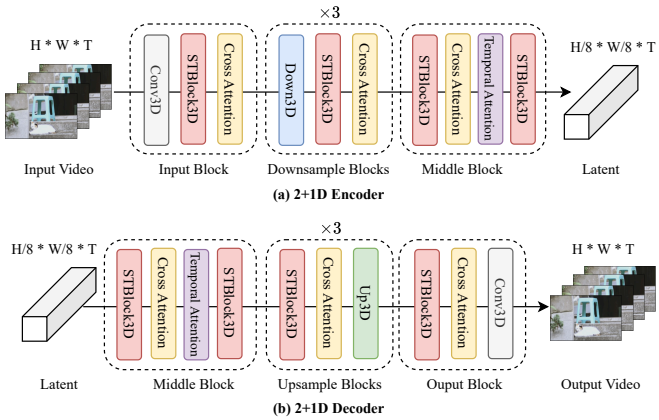


Figure 3. The architecture of our temporal-aware spatial autoencoder. We expand the 2D convolution of SD VAE [21] to 3D convolution and append one additional 3D convolution as temporal convolution after the expanded 3D convolution, which forms the STBlock3D. We also inject the cross-attention layers for cross-modal learning with textual conditions.

information at the first stage, but introduce another temporal encoder to further encode the temporal dimensions, which serves as the second stage compression. We follow the same design of option 2 for our temporal encoder and decoder. After that, we decode the reconstructed latent of the second stage to the RGB space, with the inflated 3D decoder. We jointly train the inflated 3D VAE and the temporal autoencoder. The main idea is illustrated in Fig. 2 and Fig. 3.

**Formulation** Recall $\mathbf{X} \in \mathbb{R}^{C\times T\times H\times W}$ represent a video, where $C$, $T$, $H$, and $W$ denote the number of channels, frames, height, and width, respectively. The $i$-th frame of the video is denoted as $\mathbf{x}_i \in \mathbb{R}^{C\times H\times W}$. The temporal-aware spatial encoder encodes $\mathbf{X}$ into a latent representation $\mathbf{Z}_1 \in \mathbb{R}^{c\times T\times h\times w}$, where $c$ is the number of latent channels, and $h = \frac{H}{8}$, $w = \frac{W}{8}$, as formulated by:

$$\mathbf{Z}_1 = \mathcal{E}_1(\mathbf{X}). \tag{2}$$

Next, the temporal autoencoder encodes $\mathbf{Z}_1$ into $\mathbf{Z}_2 \in \mathbb{R}^{c'\times t\times h\times w}$, where $c'$ is the number of latent channels for $\mathbf{Z}_2$ and $t = \frac{T}{4}$, as given by:

$$\mathbf{Z}_2 = \mathcal{E}_2(\mathbf{Z}_1). \tag{3}$$

Reconstruction is achieved by decoding $\mathbf{Z}_2$ back into the original video space, $\hat{\mathbf{X}} \in \mathbb{R}^{C\times T\times H\times W}$, through the following inverse process:

$$\hat{\mathbf{X}} = \mathcal{D}_1(\mathcal{D}_2(\mathbf{Z}_2)) = \mathcal{D}_1(\mathbf{Z}_1). \tag{4}$$

### 3.3. Cross-modal Modeling

Since textual information is a native component for text-to-video generation datasets, we examine if the textual information can improve the autoencoding process of the model. To achieve that, we split the feature maps into patches as tokens after each ResNet block in the encoder and decoder, and compute the cross attention by taking visual tokens as query (Q) and value (V), the text embeddings as key (K).

We try to keep the patch size trackable for each layer. Specifically, we use patch size to $8\times8$, $4\times4$, $2\times2$, and $1\times1$ for each layer in the temporal-aware spatial autoencoder respectively. We directly use each pixel as one patch in the temporal autoencoder. We adopt LayerNorm as the normalization function. We use Flan-T5 [12] as the text embedder. A projection convolution is applied to the result, which is then added to the input via a residual connection.

### 3.4. Joint Image and Video Compression

In contrast to existing architectures such as MagVitV2 [33], OD-VAE [9], and OPS-VAE [36], which use Causalconv3D layers, we rely primarily on standard Conv3D layer.

A notable feature of our architecture is the ability to mask out the temporal autoencoder, allowing the first-stage model to operate as a standalone image compressor. During training, our model is flexible to take both image and video as input: when the current batch is composed of images, we will disable the temporal convolution and temporal attention layers, as well as the temporal autoencoder. We train our model on both the image dataset and video dataset to let the model learn the image and video compression ability simultaneously. Besides, training on more high-quality images can also help improve the video autoencoding performance. We quantitatively evaluate the performance of our joint image and video compression and compare it with alternative solutions in Table 1.

### 3.5. Loss Functions

We use the reconstruction loss, the KL divergence loss, and the video adversarial loss (3D GAN loss) to optimize our model. The reconstruction loss, $\mathcal{L}_{\text{recon}}$, ensures that the generated frames are perceptually and structurally similar to the input frames. It combines a pixel-wise error term with a perceptual loss, weighted by a hyperparameter. The KL divergence loss, $\mathcal{L}_{\text{KL}}$, regularizes the latent space by encouraging it to conform to a prior distribution, ensuring smoothness and continuity in the learned latent representations. Given the hierarchical structure of our latent space, we only regularize the innermost latent $\mathbf{Z_2}$, with dimensions $\frac{T}{4}\times\frac{H}{8}\times\frac{W}{8}$, where $T$, $H$, and $W$ represent the temporal, height, and width dimensions, respectively. The 3D GAN loss, $\mathcal{L}_{\text{GAN}}$, is introduced to enhance the realism of the generated video sequences, leveraging a discriminator to distinguish between


Figure 4. Comparisons among simultaneous spatiotemporal modeling, sequential spatiotemporal modeling and our proposed solution. More results can be found in the supplement.

real and generated sequences. The total loss function is expressed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{GAN}}\mathcal{L}_{\text{GAN}}. \quad (5)$$

# 4. Experiments

## 4.1. Experimental Setup

**Training Details** We train our models on three datasets: the public Panda2M [10] with 2.28M videos, MM-Trailer [11] with 3.11M videos, and an internal text-video dataset with 0.48M videos. In total the training video number is 5.87M videos. To start training, we initialize our 4-channel and 16-channel Video VAE models from SD1.4 [21] and SD3.5 [1], respectively. For both models, we enable the video GAN loss after 50K warmup steps. We initially train the 4-channel and 16-channel latent Video VAEs for 230K and 310K steps, respectively. Subsequently, we conduct joint image-video training, empirically using an 8:2 video-to-image ratio to balance video and image reconstruction. The training images are randomly selected from our training video datasets. By masking the temporal dimension and bypassing the temporal autoencoder, we treat these images as independent static frames, allowing the model to learn from both temporal and spatial information. The 4-channel and 16-channel latent Video VAEs undergo additional joint training for 100K and 185K steps, respectively. For the cross-modal VAE, both models were initialized with their pre-trained weights. We trained the 4-channel cross-modal VAE on video-text pairs for 215,000 steps and the 16-channel variant for 180,000 steps, enabling the model to learn robust alignment between visual and textual modalities.

**Evaluation Datasets** We use three test sets for evaluation: the WebVid-10M [2] test set, the Inter4K [22] test set, and a self-constructed large motion test set. Specifically, the WebVid test set contains 1,000 videos. Each video is center cropped and resized into 256×256 resolution. The Inter4K test set consists of 500 videos with a different resolution setting of 480×864 for a comprehensive evaluation. The self-constructed large motion test set is used to better assess the model's ability to handle challenging large motion patterns. The construction details are as follows: We filter videos from WebVid-10M test set and Inter4K test set with large motions based on average optical flow scores. 80 videos with top score are selected. The final average optical flow of three evaluation sets are 12.585 (Large motion), 5.905 (WebVid), and 9.1339 (Inter4K), indicating the motion level is aligned with our expectations. We randomly select a clip of 16 consecutive frames from each video with the same frame stride as the training which is 4. The FPS of the evaluated video is the FPS of original video divide by the sampling frame stride.

**Evaluation Metrics** We use both pixel-level and perceptual-level video reconstruction metrics including PSNR, SSIM, and LPIPS [34] to quantitatively measure the reconstruction performance of video semantics and details. We also calculate the distribution metrics including FVD and KVD [26], to measure the quality and realism of the reconstructed distributions. Additionally, we conduct a user study based on MOS score to evaluate the temporal smoothness and overall video quality from real users.

| Model | Downsample Factor | #Channels | WebVid Test Set [2] | | | | Inter4K Test Set [22] | | | | Large Motion Test Set | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PSNR (↑) | SSIM (↑) | LPIPS (↓) | FVD (↓) | PSNR (↑) | SSIM (↑) | LPIPS (↓) | FVD (↓) | PSNR (↑) | SSIM (↑) | LPIPS (↓) | FVD (↓) |
| Open-Sora-Plan (OD VAE [9]) | 4x8x8 | 4 | 29.1646 | 0.8334 | 0.0789 | 40.0616 | 28.6690 | 0.8381 | 0.0906 | 38.3960 | 27.5697 | 0.8045 | 0.1065 | 219.4829 |
| IV-VAE [28] | 4x8x8 | 4 | 28.7000 | 0.8253 | 0.0832 | 38.4457 | 28.7599 | 0.8398 | 0.0906 | 28.6293 | 27.2838 | 0.8014 | 0.1053 | 197.0937 |
| Open-Sora (OPS VAE [36]) | 4x8x8 | 4 | 29.3753 | 0.8284 | 0.1240 | 90.8863 | **29.2721** | 0.8431 | 0.1316 | 55.1470 | **27.7586** | 0.8032 | 0.1540 | 355.5645 |
| CV-VAE [35] | 4x8x8 | 4 | 28.6795 | 0.8154 | 0.1072 | 70.7160 | 27.7437 | 0.8124 | 0.1284 | 50.9006 | 26.9456 | 0.7849 | 0.1411 | 293.7747 |
| VidTok [24] | 4x8x8 | 4 | 30.2517 | **0.8679** | 0.0645 | 33.9966 | 29.2400 | 0.8520 | 0.0865 | 33.6913 | 27.2922 | 0.8073 | 0.1044 | 193.3726 |
| **Video VAE w/o Joint Training (Ours)** | 4x8x8 | 4 | 30.2091 | 0.8656 | 0.0566 | 24.2982 | 28.9048 | 0.8543 | 0.0688 | 18.1236 | 27.3917 | 0.8078 | 0.0867 | 141.1166 |
| **Video VAE (Ours)** | 4x8x8 | 4 | 30.3140 | 0.8676 | 0.0538 | 22.4391 | 28.9227 | 0.8565 | 0.0665 | 17.6865 | 27.6236 | **0.8136** | 0.0841 | 138.6341 |
| **Cross-Modal VAE (Ours)** | 4x8x8 | 4 | **30.3486** | **0.8679** | **0.0530** | **18.1058** | 29.0295 | **0.8585** | **0.0649** | **15.0345** | 27.5009 | 0.8092 | **0.0820** | **122.4046** |
| Cosmos-Tokenizer [20] | 4x8x8 | 16 | 31.2545 | 0.8861 | 0.1030 | 24.7131 | 31.2002 | 0.8957 | 0.1071 | 19.3781 | 30.1619 | 0.8675 | 0.1194 | 109.3825 |
| CogVideoX-VAE [31] | 4x8x8 | 16 | 32.8940 | 0.9208 | 0.0504 | 14.0559 | 32.5122 | 0.9229 | 0.0532 | 10.8545 | 31.0906 | 0.8978 | 0.0685 | 68.6781 |
| EasyAnimate-VAE [30] | 4x8x8 | 16 | 32.1233 | 0.9085 | 0.0405 | 11.9533 | 31.5066 | 0.9048 | 0.0572 | 9.7797 | 30.5213 | 0.8846 | 0.0598 | 66.0293 |
| CV-VAE [35] | 4x8x8 | 16 | 32.2766 | 0.9080 | 0.0546 | 16.8478 | 31.6129 | 0.9060 | 0.0642 | 15.1846 | 30.7136 | 0.8868 | 0.0726 | 91.4076 |
| Open-Sora-Plan (WF-VAE-L [18]) | 4x8x8 | 16 | 33.1031 | 0.9214 | 0.0400 | 9.3192 | 32.5211 | 0.9153 | 0.0505 | 7.9032 | 31.1565 | 0.8958 | 0.0602 | 62.4018 |
| IV-VAE [28] | 4x8x8 | 16 | 33.1887 | 0.9227 | 0.0381 | 10.2930 | 33.2088 | 0.9270 | 0.0408 | 8.4002 | 31.9277 | 0.9086 | 0.0493 | 59.5411 |
| HunyuanVAE [27] | 4x8x8 | 16 | 33.4381 | 0.9271 | 0.0323 | 8.7744 | 33.2339 | 0.9272 | 0.0372 | 9.2010 | 32.1441 | 0.9118 | 0.0445 | 51.4671 |
| VidTok [24] | 4x8x8 | 16 | 34.3469 | 0.9364 | 0.0337 | 9.6649 | 33.4085 | 0.9305 | 0.0427 | 8.2432 | 31.9332 | 0.9080 | 0.0534 | 52.9496 |
| **Video VAE w/o Joint Training (Ours)** | 4x8x8 | 16 | 33.8844 | 0.9334 | 0.0344 | 8.5503 | 32.9416 | 0.9297 | 0.0409 | 8.6246 | 31.8471 | 0.9073 | 0.0499 | 46.6997 |
| **Video VAE w/ causal conv (Ours)** | 4x8x8 | 16 | 33.3031 | 0.9233 | 0.0382 | 9.7911 | 33.0645 | 0.9266 | 0.0415 | 9.5623 | 32.0313 | 0.9087 | 0.0494 | 50.4350 |
| **Video VAE (Ours)** | 4x8x8 | 16 | 34.1558 | 0.9362 | 0.0271 | 7.7195 | 33.3184 | 0.9328 | 0.0316 | **6.3432** | 32.1503 | 0.9122 | **0.0409** | **41.8797** |
| **Cross-Modal VAE (Ours)** | 4x8x8 | 16 | **34.4430** | **0.9403** | **0.0271** | **6.3560** | **33.7760** | **0.9371** | **0.0315** | 6.8278 | **32.2908** | **0.9174** | **0.0408** | 43.4941 |
| **Video VAE (Ours)** | 8x8x8 | 4 | 27.8445 | 0.8080 | 0.0760 | 35.2760 | 26.8921 | 0.8033 | 0.0953 | 34.4657 | 24.9830 | 0.7438 | 0.1214 | 217.8798 |
| **Cross-Modal VAE (Ours)** | 8x8x8 | 4 | **28.5882** | **0.8260** | **0.0712** | **34.4697** | **27.3075** | **0.8188** | **0.0883** | **28.9388** | **25.7957** | **0.7649** | **0.1112** | **202.7046** |

Table 1. Quantitative comparison with state-of-the-art methods.

| Model | #ch | Temporal Smoothness ↑ | Overall Quality ↑ |
|---|---|---|---|
| Open-Sora-Plan | 4 | 3.84 | 3.64 |
| Open-Sora | 4 | 3.16 | 2.74 |
| CV-VAE | 4 | 2.84 | 2.46 |
| **Our VAE** | 4 | **4.6** | **4.54** |
| Cosmos-Tokenizer | 16 | 3.18 | 2.54 |
| CogVideoX-VAE | 16 | 3.86 | 3.48 |
| EasyAnimate-VAE | 16 | 4.06 | 4.02 |
| CV-VAE | 16 | 3.36 | 3.00 |
| **Our VAE** | 16 | **4.5** | **4.4** |

Table 2. User study based on MOS scores for comparisons with baselines.

| Model | Factor | UCF101 | | WebVid | |
|---|---|---|---|---|---|
| | | FVD↓ | KVD↓ | FVD↓ | KVD↓ |
| Vc2 | 1x8x8 | 498.70 | 23.38 | 570.48 | 48.45 |
| Vc2 + **ours** | 4x8x8 | **479.69** | **20.18** | **529.75** | **37.39** |

Table 3. Applying our VAE on the video generation task.

### 4.2. Comparison with State-of-the-arts

We compare our Video VAE models with the state-of-the-art methods: Open-Sora-Plan [17], Open-Sora [36], CV-VAE [35], IV-VAE [28], VidTok [24] on the 4-channel setting, and Cosmos-Tokenizer [20], CogVideoX [31], EasyAnimate [30], CV-VAE [35], HunyuanVAE [27], IV-VAE [28], WF-VAE [18], VidTok [24] on the 16-channel setting.

**Quantitative Evaluation** We compare our method with more than 10 baselines on three test sets, as shown in Table 1. Among these, our 4-channel latent Video VAE demonstrates superior performance across most of datasets and metrics. Specifically, our model achieves the best reconstruction quality on the WebVid test set, shown as more than 1dB improvements over baselines and a significant improvement on the LPIPS metrics, which indicates our reconstruction is both with high-fidelity and better perceptual quality. A similar conclusion can be made on the Inter4K test set. On the large motion test set, our model maintains strong performance with a significant PSNR, SSIM, LPIPS and FVD improvement, showcasing its superiority in handling complex motion scenarios.

For models with 16-channel latent space, our model consistently outperforms these baselines across all test sets. For example, on the WebVid test set, our model achieves more than 2dB in terms of PSNR, significantly higher than Cosmos-Tokenizer and CogVideoX. Moreover, our model achieves the best SSIM and LPIPS, demonstrating substantial improvements in both fidelity and perceptual quality.

**Qualitative Evaluation** We provide qualitative comparisons with the baselines in Fig. 1. Our method demonstrates significantly improved motion recovery, greatly reducing ghosting artifacts even in rapid motion scenarios. In contrast, Open-Sora-Plan and CV-VAE struggle to reconstruct fast-moving objects, leading to ghosting artifacts. Additionally, Open-Sora VAE introduces color reconstruction errors, as seen in the clothing of the moving figure. Increasing the latent channels to 16 improves motion reconstruction across all baselines, but noticeable detail errors remain. Our 16-channel model further mitigates these errors, resulting in more accurate detail reconstruction.

**User study** We conduct a user study to further compare our method against baselines. We conduct the user study using Mean Opinion Score (MOS) [23]. We randomly sample 10 videos and ask the users to rank the videos from different methods according to two criteria: temporal smoothness and overall video quality. The results are shown in Table 2. Our participants include researchers, engineers, and university students—approximately 50% male and 50% fe-

| Model | PSNR (↑) | SSIM (↑) | LPIPS (↓) |
|---|---|---|---|
| **Simultaneous** | 24.0593 | **0.7315** | 0.1293 |
| **Sequential** | 23.3681 | 0.6917 | 0.1481 |
| **Ours** | **24.6722** | 0.7234 | **0.1162** |

Table 4. Ablation study comparing simultaneous modeling, sequential modeling, and ours on the large-motion test set.

male, mostly aged 20-30. Questionnaires were carefully designed to ensure a clear understanding of evaluation criteria. Our methods obtain the highest MOS on both the temporal smoothness and the overall video quality.

### 4.3. Compatibility with video generation models

We investigate the compatibility of our VAE's latent space with video diffusion models. For this assessment, we employ the widely recognized open-source video generation model VideoCrafter2 (VC2) [8]. Specifically, we finetuned VC2 using both our VAE and VC2's original VAE on the UCF dataset. The comparison results measured by Fréchet Video Distance (FVD) and Kernel Video Distance (KVD) are presented in Table 3. Our model achieves superior generation results while maintaining a higher compression rate. Additionally, we finetuned VC2 with our VAE on a private dataset, then evaluated both the modified and original VC2 on 1,000 WebVid videos. These comparative results are shown in Table 3. The findings further validate our VAE's effectiveness in video generation applications.

### 4.4. Ablation Study

**Spatiotemporal Compression Architecture** The visual ablation and quantitative ablation results are shown in Fig. 4 and Table 4. We compare different spatiotemporal compression strategies, including simultaneous and sequential spatiotemporal compression, and our proposed solution. These architecture variants are tested on the large motion test set to determine which model handles challenging scenarios most effectively, as shown in Table 4.

**Cross-modal guidance** We provide visual ablation results on the cross-modal training on the 4-channel model, as shown in Fig. 5. The textual guidance can improve the fine-grained detail recovery, such as the text on the man's clothes, and the eye of the woman. The quantitative ablation results are shown in Table 1, from where we can see a clear improvement between "Cross-Modal VAE (Ours)" and "Video VAE (Ours)" under all three difference settings: 4×8×8-4 channels, 4×8×8-16 channels, and 8×8×8-4 channels. Additionally, we analyse this design on the convergence of training. The figure of convergence curve during training is documented in the Section C of the supplementary materials. From the figure we can see that adopting textual guidance can speed up the converge of the model training.


Figure 5. Visual ablation of the proposed cross-modal guidance on the 4-ch models. The fine-grained reconstruction of small regions such as the definition of the text on the cloth, and the woman's eye in the second case are improved. We also visualize the learned attention map which shows the focused semantics of the video.

**Joint Training** We evaluate the effectiveness of our image-video joint training on video reconstruction on both 4-channel and 16-channel settings, as shown in Table 1 ("Video VAE w/o Joint Training (Ours)" *v.s.* "Video VAE (Ours)"). We also compare our joint training strategy with causal convolution-based methods [9, 32, 36] in Table 1 ("Video VAE w/ causal conv (Ours)"). The joint training can further boost the performance of video reconstruction in both the 4-channel and 16-channel experiments. The visual results are documented in the Section A of the supplementary materials. We also show the image reconstruction results and compare our models with SD 1.4 and SD 3.5, as shown in the Section A of the supplementary materials. Our performance is better than the SD 1.4 and almost comparable with SD 3.5. Overall, these results demonstrate that our joint image-video training strategy allows the model to retain strong image reconstruction capability while improve the performance of video reconstruction.

## 5. Conclusion

We propose VideoVAE+ to address the challenging problem of large motion video autoencoding. VideoVAE+ introduces two key innovations: an improved spatiotemporal modeling architecture and the incorporation of textual guidance. To enhance reconstruction quality and versatility, we employ joint image-video training. Extensive experiments on various datasets and settings demonstrate the superior performance of our model over state-of-the-art baselines, particularly in challenging large motion scenarios.

**Acknowlegement** This research was supported by the Research Grant Council of the Hong Kong Special Administrative Region under grant number 16212623.

## References


[1] Stability AI. Stable diffusion 3.5 large, 2023. 6

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021. 6, 7

[3] Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024. 3

[4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1, 3

[5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 1

[6] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1, 3

[7] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 1, 3

[8] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models, 2024. 1, 3, 8

[9] Liuhan Chen, Zongjian Li, Bin Lin, Bin Zhu, Qian Wang, Shenghai Yuan, Xing Zhou, Xinhua Cheng, and Li Yuan. Od-vae: An omni-dimensional video compressor for improving latent video diffusion model, 2024. 5, 7, 8

[10] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024. 6

[11] Xiaowei Chi, Yatian Wang, Aosong Cheng, Pengjun Fang, Zeyue Tian, Yingqing He, Zhaoyang Liu, Xingqun Qi, Jiahao Pan, Rongyu Zhang, Mengfei Li, Ruibin Yuan, Yanbing Jiang, Wei Xue, Wenhan Luo, Qifeng Chen, Shanghang Zhang, Qifeng Liu, and Yike Guo. Mmtrail: A multimodal trailer video dataset with language and music descriptions, 2024. 6

[12] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. 5

[13] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*. 1

[14] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022. 1

[15] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation, 2023. 3

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014. 3

[17] PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, 2024. 1, 3, 7

[18] Zongjian Li, Bin Lin, Yang Ye, Liuhan Chen, Xinhua Cheng, Shenghai Yuan, and Li Yuan. Wf-vae: Enhancing video vae by wavelet-driven energy flow for latent video diffusion model. *arXiv preprint arXiv:2411.17459*, 2024. 7

[19] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 3

[20] NVIDIA. Cosmos tokenizer: A suite of image and video neural tokenizers, 2024. 1, 7

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 3, 5, 6

[22] Alexandros Stergiou and Ronald Poppe. Adapool: Exponential adaptive pooling for information-retaining downsampling. *arXiv preprint*, 2021. 6, 7

[23] Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016. 7

[24] Anni Tang, Tianyu He, Junliang Guo, Xinle Cheng, Li Song, and Jiang Bian. Vidtok: A versatile and open-source video tokenizer. *arXiv preprint arXiv:2412.13061*, 2024. 7

[25] The Movie Gen team @ Meta. Movie gen: A cast of media foundation models. 2024. 3

[26] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019. 6

[27] Zijian Zhang Rox Min Zuozhuo Dai Jin Zhou Jiangfeng Xiong Xin Li Bo Wu Jianwei Zhang Kathrina Wu Qin Lin Aladdin Wang Andong Wang Changlin Li Duojun Huang Fang Yang Hao Tan Hongmei Wang Jacob Song Jiawang Bai Jianbing Wu Jinbao Xue Joey Wang Junkun Yuan Kai Wang Mengyang Liu Pengyu Li Shuai Li Weiyan Wang Wenqing Yu Xinchi Deng Yang Li Yanxin Long Yi Chen Yutao Cui

Yuanbo Peng Zhentao Yu Zhiyu He Zhiyong Xu Zixiang Zhou Zunnan Xu Yangyu Tao Qinglin Lu Songtao Liu Dax Zhou Hongfa Wang Yong Yang Di Wang Yuhong Liu Weijie Kong, Qi Tian and along with Caesar Zhong Jie Jiang. Hunyuanvideo: A systematic framework for large video generative models, 2024. 1, 7

[28] Pingyu Wu, Kai Zhu, Yu Liu, Liming Zhao, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Improved video vae for latent video diffusion model. *arXiv preprint arXiv:2411.06449*, 2024. 7

[29] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors, 2023. 3

[30] Jiaqi Xu, Xinyi Zou, Kunzhe Huang, Yunkuo Chen, Bo Liu, MengLi Cheng, Xing Shi, and Jun Huang. Easyanimate: A high-performance long video generation method based on transformer architecture, 2024. 1, 3, 7

[31] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer, 2024. 1, 3, 7

[32] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023. 8

[33] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *International Conference on Learning Representations*, 2024. 3, 5

[34] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[35] Sijie Zhao, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Muyao Niu, Xiaoyu Li, Wenbo Hu, and Ying Shan. Cv-vae: A compatible video vae for latent generative video models. *arXiv preprint arXiv:2405.20279*, 2024. 1, 3, 7

[36] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024. 1, 3, 5, 7, 8

[37] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 1