

# DC-VideoGen: Efficient Video Generation with Deep Compression Video Autoencoder

Junyu Chen<sup>†</sup>, Wenkun He<sup>†</sup>, Yuchao Gu<sup>†</sup>, Yuyang Zhao, Jincheng Yu, Junsong Chen, Dongyun Zou, Yujun Lin, Zhekai Zhang, Muyang Li, Haocheng Xi, Ligeng Zhu, Enze Xie, Song Han, Han Cai

NVIDIA

<sup>†</sup>Equal Contribution

<https://github.com/dc-ai-projects/DC-VideoGen>

**Abstract:** We introduce DC-VideoGen, a post-training acceleration framework for efficient video generation. DC-VideoGen can be applied to any pre-trained video diffusion model, improving efficiency by adapting it to a deep compression latent space with lightweight fine-tuning. The framework builds on two key innovations: (i) a **Deep Compression Video Autoencoder** with a novel chunk-causal temporal design that achieves  $32\times/64\times$  spatial and  $4\times$  temporal compression while preserving reconstruction quality and generalization to longer videos; and (ii) **AE-Adapt-V**, a robust adaptation strategy that enables rapid and stable transfer of pre-trained models into the new latent space. Adapting the pre-trained Wan-2.1-14B model with DC-VideoGen requires only 10 GPU days on the NVIDIA H100 GPU. The accelerated models achieve up to  $14.8\times$  lower inference latency than their base counterparts without compromising quality, and further enable  $2160\times3840$  video generation on a single GPU.

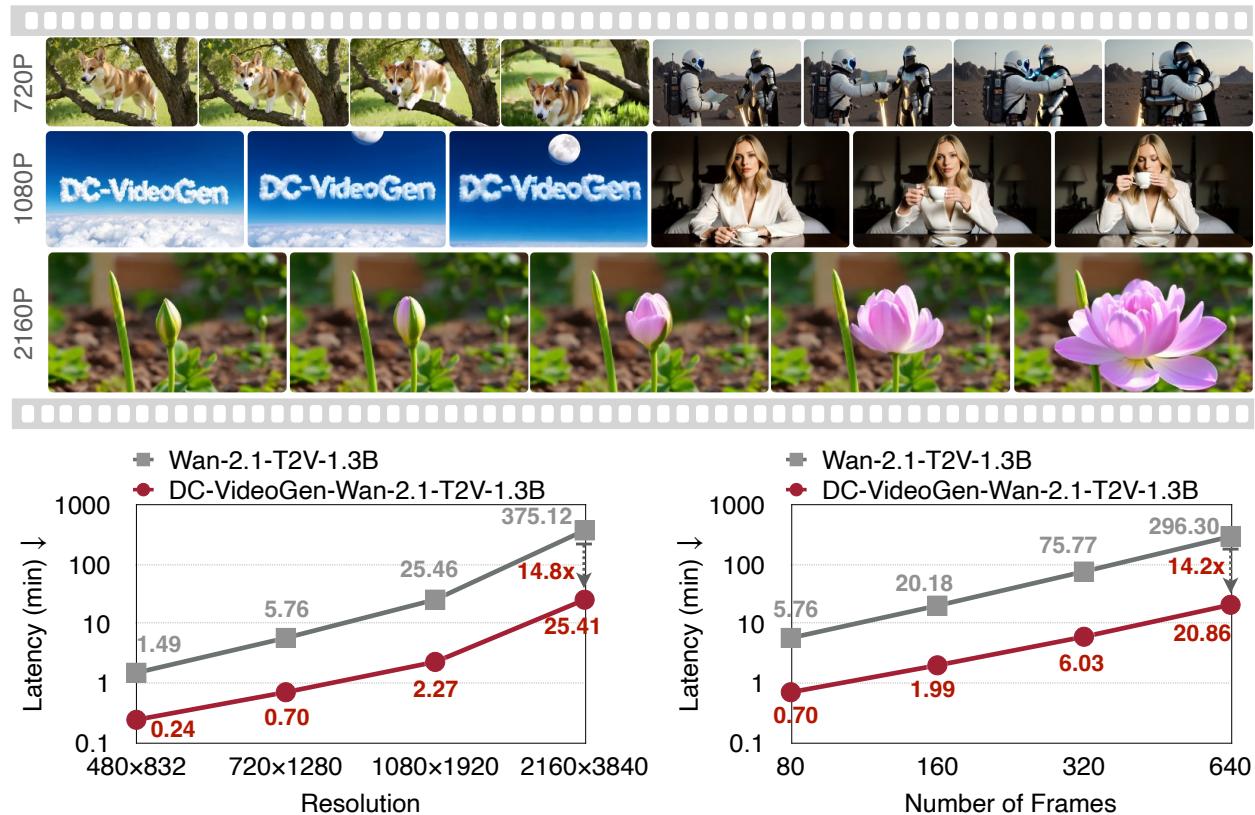


Figure 1 | DC-VideoGen can generate high-quality videos on a single NVIDIA H100 GPU with resolutions ranging from 480px, 720px, 1080px, and 2160px. On  $2160\times3840$  resolution, DC-VideoGen delivers  $14.8\times$  acceleration compared to the Wan-2.1-T2V-1.3B model.

## 1. Introduction

Video generation has rapidly become a central research focus in generative modeling, driven by its potential to enable applications in creative media, digital communication, virtual product visualization, and world simulation for autonomous driving and robotics. Recent advances in diffusion models and large-scale training have made it possible to synthesize high-quality, temporally coherent videos, substantially narrowing the gap between synthetic and real-world content [1, 2]. Industry-scale systems such as Veo3 [3], Kling [4], Wan [5], and Seedance [6] have shown that increasing model size and training data leads to significant improvements in video fidelity. Despite these advances, such models remain extremely computationally demanding in both training and inference, limiting their accessibility and practical deployment.

This paper introduces DC-VideoGen, a novel post-training framework for accelerating video diffusion models. Figure 1 showcases high-resolution video samples generated by models accelerated with DC-VideoGen. The framework supports video generation at up to  $2160 \times 3840$  resolution on a single NVIDIA H100 GPU, achieving a  $14.8 \times$  inference speedup over the base model. Moreover, DC-VideoGen dramatically reduces training costs compared with training video diffusion models from scratch. For instance, accelerating Wan-2.1-14B [5] with DC-VideoGen requires only 10 H100 GPU days —  $230 \times$  less than the training cost of Wan-2.1-14B [5]. DC-VideoGen is built upon two core innovations.

**i) Deep Compression Video Autoencoder.** Video data exhibits redundancy across both spatial and temporal dimensions [7, 8]. To mitigate training and inference costs, modern video diffusion models typically employ a video autoencoder that compresses raw videos into a more compact latent space. However, existing video autoencoders generally achieve only moderate compression ratios (e.g.,  $8 \times$  spatial and  $4 \times$  temporal), which remain insufficient for generating high-resolution or long-duration videos. In Section 3.2, we introduce the Deep Compression Video Autoencoder (**DC-AE-V**), which achieves  $32 \times / 64 \times$  spatial compression and  $4 \times$  temporal compression while preserving high reconstruction quality. The core design is a novel chunk-causal temporal modeling approach (Figure 4), which integrates bidirectional information flow within chunks and causal information flow across chunks. This design substantially improves reconstruction quality under deep compression settings (Figure 5) while preserving generalization to longer videos during inference (Figure 3).

**ii) AE-Adapt-V.** After obtaining the deep compression latent space from DC-AE-V, we introduce AE-Adapt-V in Section 3.3.2, which efficiently adapts pre-trained video diffusion models to this latent space through lightweight finetuning (Figure 2). The core design is a video embedding space alignment stage (Figure 7), which helps recover the base model’s knowledge and semantics in the new latent space by aligning the patch embedder and output head. These aligned modules provide a robust initialization, enabling rapid recovery of the base model’s quality (Figure 6) through LoRA finetuning (Figure 8).

Extensive evaluations on video reconstruction (Table 1), text-to-video generation (Tables 2, 3), and image-to-video generation (Table 4) demonstrate the effectiveness of DC-VideoGen. Across tasks, it consistently provides substantial efficiency gains while achieving comparable or even superior VBench scores. We summarize our main contributions below:

- We introduce DC-VideoGen, a general framework for accelerating video diffusion models. With low-cost post-training finetuning, it delivers substantial efficiency gains in video generation.
- We introduce DC-AE-V, which drastically reduces the number of latent space tokens while preserving high reconstruction quality and generalization to longer videos.
- We introduce AE-Adapt-V, which enables rapid adaptation of pre-trained diffusion models to the latent spaces of new autoencoders.
- DC-VideoGen provides accelerated video diffusion models that preserve the quality of the base models while achieving exceptional efficiency, supporting video generation at up to  $2160 \times 3840$  resolution on a single GPU. This offers practical advantages for applications requiring efficient video synthesis. Moreover, our accelerated models incur lower fine-tuning and training costs than their base counterparts, enabling faster innovation in the video generation community.

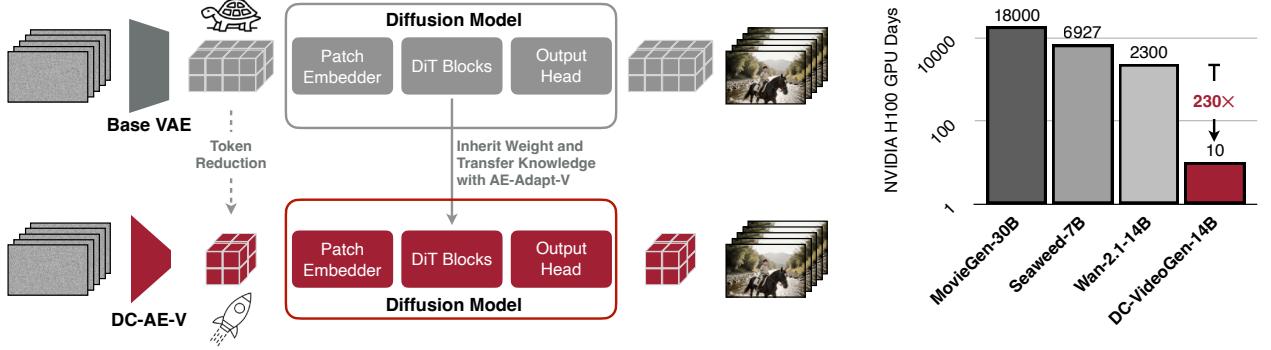


Figure 2 | **DC-VideoGen Overview.** DC-VideoGen is a post-training acceleration framework for video diffusion models. It achieves acceleration by transferring models to DC-AE-V’s latent space (Section 3.2) and rapidly recovering the base model’s quality and semantics using AE-Adapt-V (Section 3.3.2). Compared with training from scratch, DC-VideoGen is far more efficient — for example, DC-VideoGen-14B requires only 10 NVIDIA H100 GPU days, a  $230\times$  reduction in training cost relative to Wan-2.1-14B.

## 2. Related Work

**Video Autoencoder.** To circumvent the prohibitive costs of training and running diffusion models in pixel space, latent video diffusion models commonly employ video autoencoders [9, 10, 11, 12] to compress raw videos into a compact latent space, enabling more efficient generation. A typical configuration in recent works [13, 14, 15, 16, 17, 18, 19, 20, 21, 5] uses  $8\times$  spatial and  $4\times$  temporal compression. Some studies [2, 5, 22] explore  $16\times$  spatial compression to further reduce latent token counts. However, these configurations are often insufficient for high-resolution or long video generation. Inspired by the success of  $32\times$  spatial compression in image autoencoders [23], [24, 25] investigate video autoencoders with a  $32\times$  spatial compression ratio, but they suffer from low reconstruction quality or poor generalization to longer videos. In contrast, our DC-AE-V achieves up to  $64\times$  spatial compression while maintaining superior reconstruction quality and generalization.

**Efficient Autoencoder Adaptation for Video Diffusion Models.** Closely related to our work, OpenSora 2.0 [25] explored adapting pre-trained video diffusion models to their autoencoders by directly loading the pretrained DiT backbone weights while randomly initializing the patch embedder and output head. Their experiments show that this approach produces noticeably blurry videos and fails to match the performance of training from scratch, a finding consistent with our experiments (Figure 6). To address this challenge, we introduce a video embedding space alignment stage that recovers the base model’s knowledge in the new latent space.

**Video Diffusion Model Acceleration.** To accelerate video diffusion models, one line of research focuses on reducing the number of diffusion steps [26, 27, 28, 29, 30, 31, 32, 33, 34]. Another line explores model compression, including sparsity [7, 8, 35, 36, 37, 38, 39, 40] and quantization [41, 42, 43, 44, 45, 46, 40, 47]. Our DC-VideoGen is complementary to them, as it accelerates video generation by reducing token redundancy.

## 3. Method

### 3.1. DC-VideoGen Overview

Generating high-resolution or long videos with video diffusion models is computationally expensive due to the large number of latent tokens. Furthermore, the prohibitive pre-training costs make developing new video diffusion models both challenging and risky [48].

This paper addresses these challenges from two complementary perspectives (Figure 2, left). First, we drastically reduce the number of tokens using our deep compression video autoencoder. Second, we introduce a cost-efficient post-training strategy to adapt pre-trained models to new autoencoders. This approach substantially lowers the risk, training cost, and reliance on large high-quality datasets. As shown in Figure 2 (right), applying our post-training strategy to Wan-2.1-14B [5] takes 10 H100 GPU days — just 0.05% of the training cost of MovieGen-30B [49].

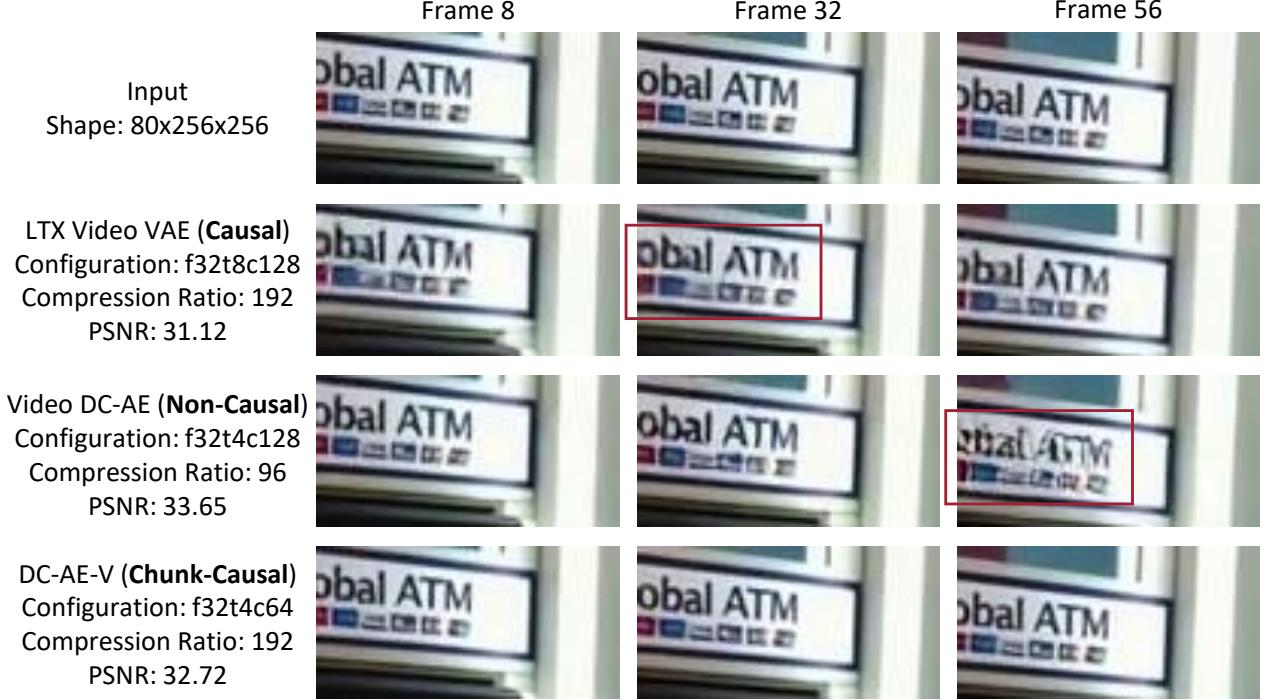


Figure 3 | **Video Autoencoder Reconstruction Visualization.** Under deep compression settings, causal video autoencoders suffer from low reconstruction quality. In contrast, non-causal video autoencoders achieve better reconstruction quality but generalize poorly to longer videos.

### 3.1.1. Preliminaries and Notation

We use  $\text{fxtycz}$  to denote the configuration of a video autoencoder. For example, **f8t4c16** represents a video autoencoder that compresses an input video of size  $3 \times T \times H \times W$  into a latent of size  $16 \times \frac{T}{4} \times \frac{H}{8} \times \frac{W}{8}$ . The compression ratio is defined as

$$\text{Compression Ratio} = \frac{3THW}{c \cdot \frac{T}{t} \cdot \frac{H}{f} \cdot \frac{W}{f}} = \frac{3f^2t}{c}. \quad (1)$$

Given the same reconstruction quality, a higher compression ratio is generally preferred [50]. We refer to diffusion models with an  $\text{fxtycz}$  autoencoder as an “**fxtycz model**”.

A video diffusion model typically comprises a single-layer patch embedder that maps the latent space to the embedding space, transformer blocks, and an output head that projects back to the latent space (Figure 7c). The patch embedder includes a hyperparameter called the patch size  $p$ , which further spatially compresses the latent by a factor of  $p \times$ . As shown in [23], for the same total compression ratio, allocating more spatial compression to the autoencoder rather than the patch embedder yields better generation results.

## 3.2. Deep Compression Video Autoencoder

Existing video autoencoders can be categorized into two groups based on their temporal modeling design: causal and non-causal.

- **Causal.** In causal video autoencoders, information flows only from earlier frames to later frames (Figure 4b). This design naturally supports longer videos during inference, since the encoding and decoding of later frames do not affect earlier ones. However, because each frame can only leverage redundancy from preceding frames, reconstruction accuracy is limited under deep compression settings, as illustrated in Figure 3. Building on the causal design, IV-VAE [20] notes that, since every  $t$  input frames are compressed into a single latent frame, enforcing causality within each group of  $t$  frames is unnecessary. To address this, IV-VAE introduces a grouped causal convolution with group size  $t$  to improve reconstruction performance.

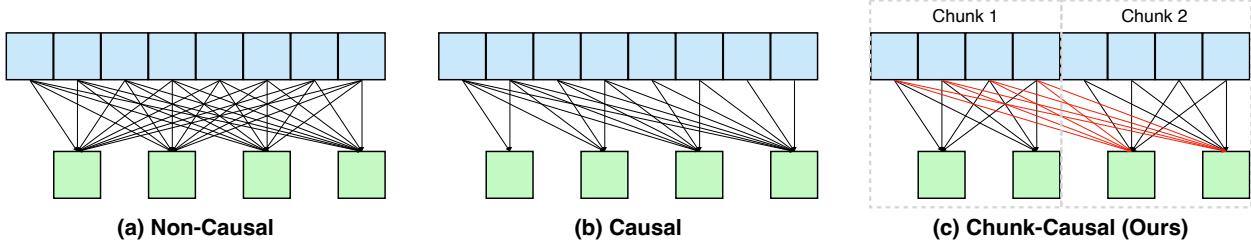


Figure 4 | **Illustration of Chunk-Causal Temporal Modeling in DC-AE-V.** Our chunk-causal temporal modeling preserves causal information flow across chunks while enabling bidirectional flow within each chunk. This design improves reconstruction quality over non-causal temporal modeling, while maintaining generalization to longer videos at inference time.

Video Autoencoder	Config	Compress. Ratio	Panda70m			UCF101		
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
VideoVAEPlus [21]	f8t4c16	48	36.88	0.968	0.009	35.79	0.959	0.016
CogVideoX VAE [18]	f8t4c16	48	35.54	0.961	0.021	34.53	0.949	0.034
HunyuanVideo VAE [19]	f8t4c16	48	35.46	0.960	0.015	34.40	0.950	0.024
IV VAE [20]	f8t4c16	48	35.32	0.959	0.017	34.84	0.955	0.025
Wan 2.1 VAE [5]	f8t4c16	48	34.15	0.952	0.017	33.81	0.943	0.024
Wan 2.2 VAE [5]	f16t4c48	64	35.12	0.958	0.013	34.27	0.948	0.022
StepVideo VAE [22]	f16t8c64	96	32.17	0.930	0.043	32.17	0.930	0.043
Video DC-AE <sup>†</sup> <sub>tiling &amp; blending</sub> [25]	f32t4c128	96	34.10	0.952	0.023	33.65	0.945	0.034
Video DC-AE [25]	f32t4c128	96	31.73	0.915	0.040	31.52	0.914	0.047
LTX Video VAE [24]	f32t8c128	192	32.41	0.928	0.039	31.12	0.910	0.059
DC-AE-V	f32t4c256	48	39.56	0.979	0.008	37.14	0.967	0.018
	f32t4c128	96	37.37	0.968	0.013	34.83	0.951	0.026
	f32t4c64	192	35.03	0.953	0.019	32.71	0.931	0.035
	f32t4c32	384	33.07	0.933	0.027	30.83	0.909	0.046
	f64t4c128	384	32.79	0.932	0.030	30.60	0.907	0.048

Table 1 | **Video Autoencoder Reconstruction Results.** <sup>†</sup>Video DC-AE achieves higher PSNR with tiling and blending, but still exhibits poor generalization when applied to longer videos at inference time, as shown in Figure 3.

However, the group size is strictly tied to the temporal compression ratio, and as shown in Figure 5, it provides only limited improvements in reconstruction quality over the standard causal design under deep compression settings.

- **Non-Causal.** Non-causal autoencoders allow bidirectional information flow between frames (Figure 4a). This enables each frame to leverage redundancy from both past and future frames, yielding better reconstruction quality under deep compression settings. However, because earlier frames depend on later ones, generalization to longer videos becomes challenging. Techniques such as temporal tiling and blending [25] can partially alleviate this issue but often introduce artifacts, including temporal flickering and boundary blurring, as shown in Figure 3.

We introduce a new temporal modeling design, **chunk-causal**, to overcome these limitations (Figure 4c). The key idea is to divide the input video into fixed-size chunks, where the chunk size is treated as an independent hyperparameter. Within each chunk, we apply bidirectional temporal modeling to fully exploit redundancy across frames. Across chunks, however, we enforce causal flow so that the model can effectively generalize to longer videos at inference time. Figure 5 presents the ablation study on chunk size. We observe that increasing the chunk size consistently improves reconstruction quality. In our final design, we adopt a chunk size of 40, as the benefits plateau beyond this point while training costs continue to rise.

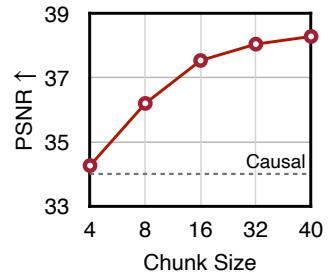


Figure 5 | **Ablation Study on Chunk Size.**

Text-to-Video Generation Results on VBench 480×832						
Text-to-Video Diffusion Model	Video Autoencoder	Patch Size	Latency (s) ↓	Score ↑		
		Overall	Quality	Semantic		
Wan-2.1-1.3B [5]	Wan-2.1-VAE-f8t4c16 [5]	2	89.30	83.32	85.01	76.57
	LTX-Video-f32t8c128 [24]	1	6.98	83.30	85.04	76.34
	OpenSora2-f32t4c128 [25]	1	14.57	82.27	84.53	73.24
	Wan-2.2-VAE-f16t4c48 [5]	2	14.59	80.38	83.20	69.08
	DC-AE-V-f64t4c128	1	3.97	83.38	85.13	76.38
	DC-AE-V-f32t4c32	1	14.55	84.48	86.02	78.33

Table 2 | Comparison of Video Generation Results across Autoencoders. We adopt the same training setup for all models to ensure apples-to-apples comparisons, i.e., using AE-Adapt-V (Section 3.3.2) to adapt the pretrained Wan-2.1-T2V-1.3B to different autoencoders.

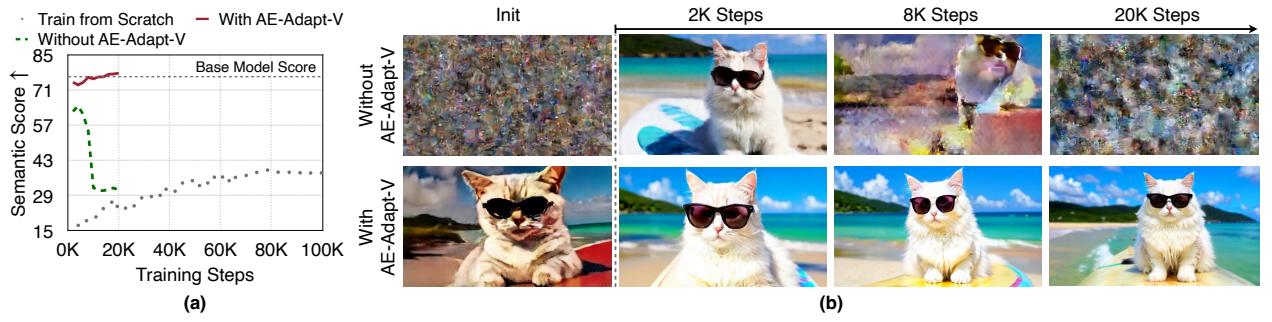


Figure 6 | Direct fine-tuning without AE-Adapt-V leads to training instability and suboptimal quality. In contrast, AE-Adapt-V provides a robust initialization that preserves semantics in the new latent space for the video diffusion model, enabling rapid recovery of visual quality and allowing the model to match the base model’s performance with lightweight fine-tuning.

**Video Reconstruction Results.** We summarize the comparison between DC-AE-V and prior state-of-the-art video autoencoders in Table 1. Compared with causal video autoencoders such as LTX Video VAE [24], DC-AE-V achieves higher reconstruction accuracy at the same compression ratio, as well as higher compression ratios for a given accuracy target. Compared with non-causal video autoencoders such as Video DC-AE [25], DC-AE-V delivers better reconstruction quality under the same compression ratio while also generalizing better to longer videos (Figure 3).

**Video Generation Results.** In addition to reconstruction performance, we also evaluate DC-AE-V against prior autoencoders on video generation. Table 2 reports ablation results on Wan-2.1-1.3B [5], showing that DC-AE-V achieves the best video generation performance. Compared with the base model, DC-AE-V-f64t4c128 provides a 22× speedup while attaining slightly higher VBench scores.

### 3.3. Post-Training Video Autoencoder Adaptation

#### 3.3.1. Naïve Approach, Challenge and Analysis

As discussed in Section 3.1.1, the patch embedder and output head are inherently tied to the latent space representation and thus cannot be transferred when replacing the autoencoder. Consequently, a straightforward approach for adapting pre-trained video diffusion models to new autoencoders is to retain the pre-trained DiT blocks while randomly initializing the patch embedder and output head (Figure 7c, right). This strategy was explored in [25], where it was found to yield unsatisfactory results.

We evaluate this approach under our settings and observe similar outcomes. As shown in Figure 6a (green dashed line), it fails to match the base model’s semantic score. Furthermore, we observe training instability:

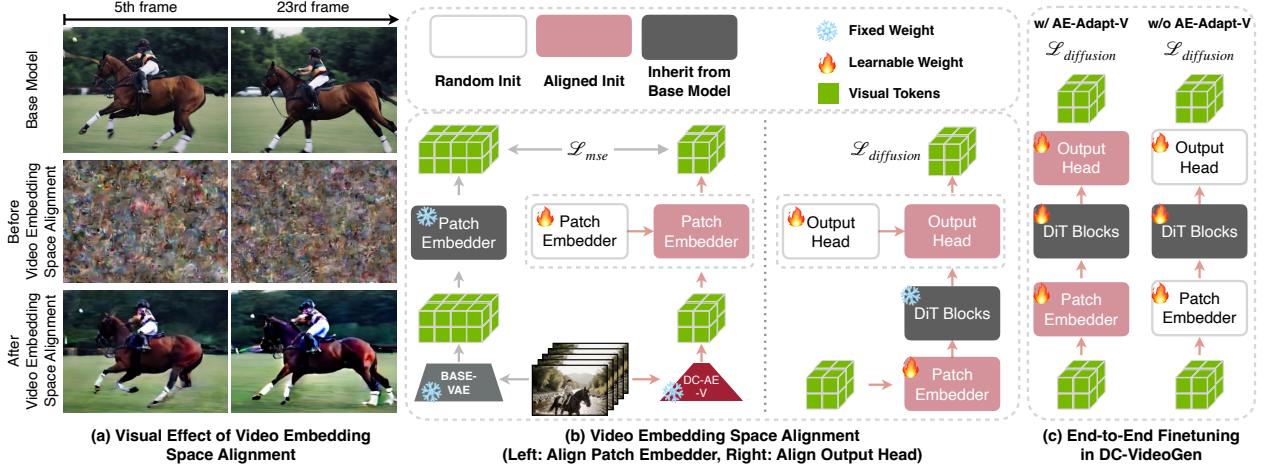


Figure 7 | **Illustration of Video Embedding Space Alignment.** We present detailed ablation studies in Figure 11, showing that both alignment steps—patch embedder alignment and output head alignment—are essential for effective video embedding space alignment.

the model’s output degrades to random noise after 20K training steps (Figure 6b, top). We conjecture that this instability arises from the substantial embedding space gap introduced by the new latent space and the randomly initialized patch embedder, which prevents the model from effectively retaining knowledge from the pre-trained DiT weights.

### 3.3.2. Our Solution: AE-Adapt-V

To address this challenge, we introduce a video embedding space alignment stage prior to end-to-end finetuning, bridging the gap between embedding spaces and preserving the pre-trained model’s knowledge while adapting to a new latent space.

**AE-Adapt-V Stage 1: Video Embedding Space Alignment.** Figure 7b illustrates the general concept of our video embedding space alignment, where we first align the patch embedder and then align the output head.

For patch embedder alignment, we freeze the base model’s patch embedder and train a new patch embedder to map from the new latent space to the embedding space. The objective is to minimize the distance between the base model’s embeddings and the embeddings produced by the new patch embedder. Formally, let the embedding of the base model be denoted as  $e_b$  with shape  $H_b \times W_b \times D$ , and the embedding of the new model as  $e_n$  with shape  $H_n \times W_n \times D$ , where  $D$  is the embedding channel dimension and  $H_n < H_b$ ,  $W_n < W_b$  in our settings. We first spatially downsample  $e_b$  using average pooling to match the shape of  $e_n$ , denoting the result as  $e'_b$ . The randomly initialized patch embedder is then trained to minimize the following loss function:

$$\mathcal{L} = \text{MSELoss}(e_n, e'_b). \quad (2)$$

With the aligned patch embedder, the output head is then aligned by jointly finetuning it and the patch embedder using the diffusion loss, while keeping the DiT blocks frozen. This process stops once the diffusion loss converges, which takes up to 4K steps in our experiments.

Figure 7a illustrates the visual effect of our video embedding space alignment. Using the aligned patch embedder and output head, we can recover the knowledge and semantics of the base model in the new latent space without updating the DiT blocks. Additional ablation studies are provided in Figure 11, which show that aligning the patch embedder plays the most critical role in video embedding space alignment, while aligning the output head further enhances the quality.

Text-to-Video Generation Results on VBench 480×832						
Text-to-Video Diffusion Model	Method	Trainable	Score ↑			
		Params (M)	Overall	Quality	Semantic	
Wan-2.1-1.3B [5]	Full-Tune	1418.90	79.81	84.02	62.98	Full-Tune
	LoRA-Tune	350.37	<b>84.48</b>	<b>86.02</b>	<b>78.33</b>	LoRA-Tune

(a) Quantitative Comparison



(b) Visual Comparison

Figure 8 | **Ablation Study on End-to-End Tuning Strategies.** LoRA attains higher scores than full fine-tuning while requiring far fewer trainable parameters.

**AE-Adapt-V Stage 2: End-to-End Fine-Tuning with LoRA.** Video embedding space alignment alone cannot fully match the base model’s quality. To close this gap, we perform end-to-end finetuning. Since stage 1 provides a strong initialization, we employ LoRA [51] tuning during this stage.

Figure 8 compares LoRA tuning with full finetuning. We find that LoRA not only reduces training cost by requiring fewer trainable parameters, but also achieves higher VBench scores and improved visual quality compared with full finetuning. We conjecture that this is because LoRA better preserves the knowledge of the base model.

### 3.4. DC-VideoGen Application

DC-VideoGen can be applied to any pre-trained video diffusion model. In our experiments, we evaluate it on two representative video generation tasks: text-to-video (T2V) and image-to-video (I2V) generation. We use pre-trained Wan-2.1 models [5] as our base models, and denote the resulting accelerated models as DC-VideoGen-Wan-2.1.

The Wan-2.1-I2V models incorporate the image condition by concatenating it with the latent. Since Wan-2.1-VAE and DC-AE-V employ different temporal modeling designs (causal vs. chunk-causal), DC-VideoGen-Wan-2.1 I2V models cannot directly adopt the same approach as Section 3.3.2. To address this, we replicate the given image condition four times and append blank frames to form chunks matching the shape of the video. We then encode these chunks with DC-AE-V and concatenate the resulting features with the latent, which can subsequently be processed in the same manner as in Wan-2.1-I2V.

## 4. Experiments

### 4.1. Setups

**Implementation Details.** We implement and train all models using PyTorch 2 [52] on 16 NVIDIA H100 GPUs. Three pretrained video diffusion models are employed: Wan-2.1-T2V-1.3B, Wan-2.1-T2V-14B, and Wan-2.1-I2V-14B, each adapted from the original Wan-2.1-VAE to our DC-AE-V. For training, we collected 257K synthetic videos using Wan-2.1-T2V-14B and combined them with 160K high-resolution videos selected from Pexels<sup>1</sup>. Detailed training hyperparameters are provided in Table 8.

**Efficiency Testbed.** We benchmark the inference latency of all models using TensorRT<sup>2</sup> on a single H100 GPU. For simplicity, we focus exclusively on the transformer backbone, as it constitutes the primary efficiency bottleneck.

**Evaluation Metrics.** Following common practice, we use VBench [53] to evaluate text-to-video (T2V) diffusion models and VBench 2.0 [54] for image-to-video (I2V) diffusion models. In addition, we provide visual results generated by our models.

<sup>1</sup><https://www.pexels.com/videos/>

<sup>2</sup><https://github.com/NVIDIA/TensorRT>

Text-to-Video Generation Results on VBench 720×1280						
Text-to-Video Diffusion Model	Video Autoencoder	#Params (B)	Latency (min) ↓	Score ↑		
				Overall Quality	Semantic	
MAGI-1 [55]	-	4.5	21.22	79.18	82.04	67.74
Step-Video [22]	-	30	13.16	81.83	84.46	71.28
CogVideoX1.5 [18]	-	5	6.73	82.17	82.78	79.76
Skyreels-V2 [56]	-	1.3	9.48	82.67	84.70	74.53
HunyuanVideo [19]	-	13	30.35	83.24	85.09	75.82
OpenSora-2.0 [25]	-	14	32.83	84.34	85.40	80.12
Wan-2.1-1.3B <sup>†</sup> [5]	Wan-2.1-VAE-f8t4c16	1.3	5.76	83.38	85.67	74.22
DC-VideoGen-Wan-2.1-1.3B	DC-AE-V-f32t4c32	1.3	0.70	84.63	86.67	76.48
Wan-2.1-14B [5]	Wan-2.1-VAE-f8t4c16	14	27.52	83.73	85.77	75.58
DC-VideoGen-Wan-2.1-14B	DC-AE-V-f32t4c32	14	3.58	84.83	86.80	76.93

Table 3 | **Results on Text-to-Video Generation.** <sup>†</sup>Native Wan-2.1-T2V-1.3B is limited to 480×832 resolution, so we fine-tune it on our dataset to support 720×1280 generation.

Image-to-Video Generation Results on VBench 720×1280						
Image-to-Video Diffusion Model	Video Autoencoder	#Params (B)	Latency (min) ↓	Score ↑		
				Overall Quality	Semantic	I2V
CogVideoX-5b-I2V [18]	-	5	6.72	86.70	78.61	94.79
HunyuanVideo-I2V [19]	-	13	30.39	86.82	78.54	95.10
Step-Video-TI2V [22]	-	30	13.18	88.36	81.22	95.50
MAGI-1 [55]	-	4.5	21.25	89.28	82.44	96.12
Wan-2.1-14B [5]	Wan-2.1-VAE-f8t4c16	14	27.88	86.86	80.83	92.90
DC-VideoGen-Wan-2.1-14B	DC-AE-V-f32t4c32	14	3.67	87.73	81.39	94.08

Table 4 | **Results on Image-to-Video Generation.**

#### 4.2. Text-to-Video Generation

Table 3 compares DC-VideoGen with leading T2V diffusion models on VBench at 720×1280 resolution. We follow the extended prompt sets provided by the VBench team and conduct all experiments at the same resolution to ensure fair, apples-to-apples comparisons.

Compared with the base Wan-2.1 models, DC-VideoGen-Wan-2.1 achieves higher scores while being significantly more efficient. For example, DC-VideoGen-Wan-2.1-14B reduces latency by 7.7× and improves the VBench score from 83.73 to 84.83 relative to Wan-2.1-14B. Compared with other T2V diffusion models, DC-VideoGen-Wan-2.1 achieves both the highest VBench score and the lowest latency. Video samples can be found in Figure 13 and the webpage in supplementary material.

#### 4.3. Image-to-Video Generation

Table 4 reports our results on VBench I2V at 720×1280 resolution. Consistent with the T2V findings, DC-VideoGen-Wan-2.1-14B outperforms the base Wan-2.1-14B by achieving a higher VBench score while reducing latency by 7.6×.

Compared with other I2V diffusion models, DC-VideoGen-Wan-2.1-14B provides highly competitive results with exceptional efficiency, running 5.8× faster than MAGI-1 and 8.3× faster than HunyuanVideo-I2V. Video samples can be found in Figure 12 and the webpage in supplementary material.

## 5. Conclusion

We introduce DC-VideoGen, a post-training framework that accelerates video diffusion models by combining a deep compression video autoencoder with an efficient adaptation strategy. DC-VideoGen achieves up to 14.8 $\times$  faster inference and drastically reduces training costs, while preserving or even improving video quality. These findings highlight that efficiency and fidelity in video generation can advance together, making large-scale video synthesis more practical and accessible for both research and real-world applications. We further discuss potential extensions and current limitations of our framework in Section A.8.

## References

- [1] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.
- [2] NVIDIA. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.
- [3] Google DeepMind. Veo 3. <https://deepmind.google/models/veo/>, 2025.
- [4] Kuaishou. Kling ai. <https://klingai.kuaishou.com/>, 2024.
- [5] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.
- [6] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.
- [7] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, et al. Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. *arXiv preprint arXiv:2502.01776*, 2025.
- [8] Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, et al. Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. *arXiv preprint arXiv:2505.18875*, 2025.
- [9] 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.
- [10] 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.
- [11] Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2535–2545, 2024.
- [12] 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. *arXiv preprint arXiv:2405.18991*, 2024.
- [13] 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. *arXiv preprint arXiv:2412.20404*, 2024.

- [14] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.
- [15] 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. *Advances in Neural Information Processing Systems*, 37:12847–12871, 2024.
- [16] 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. *arXiv preprint arXiv:2409.01199*, 2024.
- [17] Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *Advances in Neural Information Processing Systems*, 37:28281–28295, 2024.
- [18] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.
- [19] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuandvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- [20] Pingyu Wu, Kai Zhu, Yu Liu, Liming Zhao, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Improved video vae for latent video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18124–18133, 2025.
- [21] Yazhou Xing, Yang Fei, Yingqing He, Jingye Chen, Jiaxin Xie, Xiaowei Chi, and Qifeng Chen. Large motion video autoencoding with cross-modal video vae. *arXiv preprint arXiv:2412.17805*, 2024.
- [22] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025.
- [23] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024.
- [24] Yoav HaCohen, Nisan Chirrut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.
- [25] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, et al. Open-sora 2.0: Training a commercial-level video generation model in \$200 k. *arXiv preprint arXiv:2503.09642*, 2025.
- [26] Xiang Wang, Shiwei Zhang, Han Zhang, Yu Liu, Yingya Zhang, Changxin Gao, and Nong Sang. Videolcm: Video latent consistency model. *arXiv preprint arXiv:2312.09109*, 2023.
- [27] Shanchuan Lin and Xiao Yang. Animatediff-lightning: Cross-model diffusion distillation. *arXiv preprint arXiv:2403.12706*, 2024.
- [28] Jiachen Li, Weixi Feng, Tsu-Jui Fu, Xinyi Wang, Sugato Basu, Wenhui Chen, and William Yang Wang. T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. *Advances in neural information processing systems*, 37:75692–75726, 2024.
- [29] Yuanhao Zhai, Kevin Lin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Chung-Ching Lin, David Doermann, Junsong Yuan, and Lijuan Wang. Motion consistency model: Accelerating video diffusion with disentangled motion-appearance distillation. *Advances in Neural Information Processing Systems*, 37:111000–111021, 2024.
- [30] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast causal video generators. *arXiv e-prints*, pages arXiv–2412, 2024.
- [31] Fu-Yun Wang, Zhaoyang Huang, Xiaoyu Shi, Weikang Bian, Guanglu Song, Yu Liu, and Hongsheng Li. Animatelcm: Accelerating the animation of personalized diffusion models and adapters with decoupled consistency learning. *CoRR*, 2024.

- [32] Zhixing Zhang, Yanyu Li, Yushu Wu, Anil Kag, Ivan Skorokhodov, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Dimitris Metaxas, Sergey Tulyakov, et al. Sf-v: Single forward video generation model. *Advances in Neural Information Processing Systems*, 37:103599–103618, 2024.
- [33] Xiaofeng Mao, Zhengkai Jiang, Fu-Yun Wang, Jiangning Zhang, Hao Chen, Mingmin Chi, Yabiao Wang, and Wenhao Luo. Osv: One step is enough for high-quality image to video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12585–12594, 2025.
- [34] Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025.
- [35] Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. *arXiv preprint arXiv:2502.04507*, 2025.
- [36] Yifei Xia, Suhan Ling, Fangcheng Fu, Yujie Wang, Huixia Li, Xuefeng Xiao, and Bin Cui. Training-free and adaptive sparse attention for efficient long video generation. *arXiv preprint arXiv:2502.21079*, 2025.
- [37] Xin Tan, Yuetao Chen, Yimin Jiang, Xing Chen, Kun Yan, Nan Duan, Yibo Zhu, Dixin Jiang, and Hong Xu. Dsv: Exploiting dynamic sparsity to accelerate large-scale video dit training. *arXiv preprint arXiv:2502.07590*, 2025.
- [38] Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial attention:  $\mathcal{O}(n \log n)$  sparse attention with energy decay for long video generation. *arXiv preprint arXiv:2506.19852*, 2025.
- [39] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025.
- [40] Tianchen Zhao, Ke Hong, Xinhao Yang, Xuefeng Xiao, Huixia Li, Feng Ling, Ruiqi Xie, Siqi Chen, Hongyu Zhu, Yichong Zhang, et al. Paroattention: Pattern-aware reordering for efficient sparse and quantized attention in visual generation models. *arXiv preprint arXiv:2506.16054*, 2025.
- [41] Jintao Zhang, Jia Wei, Haofeng Huang, Pengle Zhang, Jun Zhu, and Jianfei Chen. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. *arXiv preprint arXiv:2410.02967*, 2024.
- [42] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. *arXiv preprint arXiv:2411.10958*, 2024.
- [43] Shilong Tian, Hong Chen, Chengtao Lv, Yu Liu, Jinyang Guo, Xianglong Liu, Shengxi Li, Hao Yang, and Tao Xie. Qvd: Post-training quantization for video diffusion models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 10572–10581, 2024.
- [44] Tianchen Zhao, Tongcheng Fang, Enshu Liu, Wan Rui, Widya Dewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, et al. Vudit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2406.02540*, 2024.
- [45] Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28306–28315, 2025.
- [46] Yushi Huang, Ruihao Gong, Jing Liu, Yifu Ding, Chengtao Lv, Haotong Qin, and Jun Zhang. Qvgen: Pushing the limit of quantized video generative models. *arXiv preprint arXiv:2505.11497*, 2025.
- [47] Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [48] Yuxian Gu, Qinghao Hu, Shang Yang, Haocheng Xi, Junyu Chen, Song Han, and Han Cai. Jet-nemotron: Efficient language model with post neural architecture search. *arXiv preprint arXiv:2508.15884*, 2025.
- [49] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.
- [50] Junyu Chen, Dongyun Zou, Wenkun He, Junsong Chen, Enze Xie, Song Han, and Han Cai. Dc-ae 1.5: Accelerating diffusion model convergence with structured latent space. *arXiv preprint arXiv:2508.00413*, 2025.

- [51] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- [52] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 929–947, 2024.
- [53] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [54] Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.
- [55] Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.
- [56] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Juncheng Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengchen Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.
- [57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [58] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakkko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.
- [59] 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.
- [60] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.
- [61] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.
- [62] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kuntschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.
- [63] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.
- [64] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016.
- [65] Shifeng Zhang, Yiliang Xie, Jun Wan, Hansheng Xia, Stan Z Li, and Guodong Guo. Widerperson: A diverse dataset for dense pedestrian detection in the wild. *IEEE Transactions on Multimedia*, 22(2):380–393, 2019.
- [66] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *European conference on computer vision*, pages 742–758. Springer, 2020.
- [67] Unsplash. Unsplash lite dataset 1.3.0, 2020.
- [68] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

- [69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [70] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.
- [71] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [72] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.
- [73] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.
- [74] Yuchao Gu, Weijia Mao, and Mike Zheng Shou. Long-context autoregressive video modeling with next-frame prediction. *arXiv preprint arXiv:2503.19325*, 2025.
- [75] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation. 2025.

## A. Appendix

### A.1. The Use of LLM

The use of LLMs was strictly limited to the final manuscript writing stage, specifically for the purpose of correcting grammatical errors and polishing the text.

### A.2. Additional Details of DC-AE-V

#### A.2.1. Model Architecture

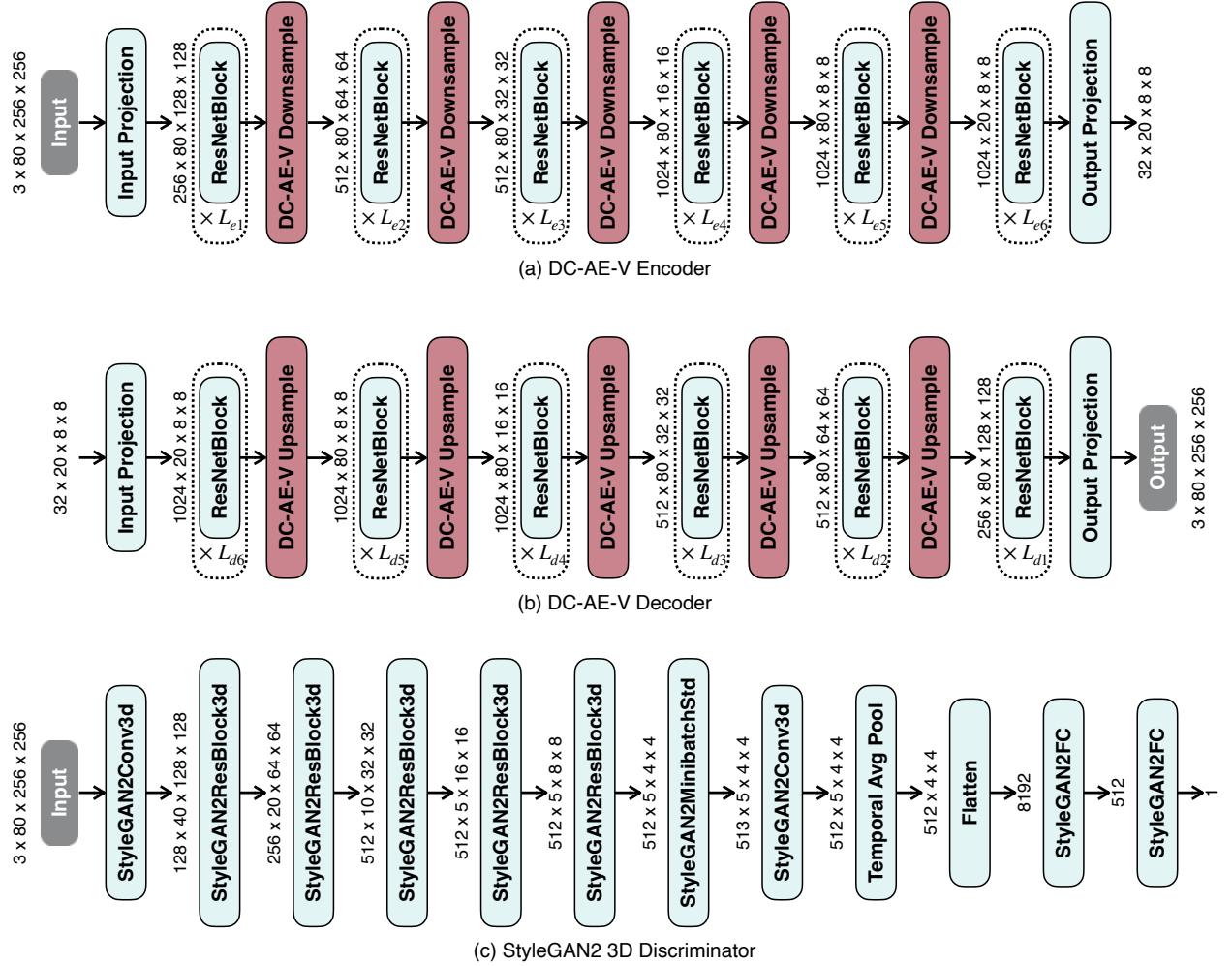


Figure 9 | Model Architecture of DC-AE-V.

Figure 9 presents the detailed model architecture of an f32t4c32 DC-AE-V. Both the encoder and decoder are composed of six stages, each built from 3D ResNet [57] blocks. The first five stages perform only spatial downsampling and upsampling, while the final stage handles temporal downsampling and upsampling. Following DC-AE [23], we incorporate Residual Autoencoding to facilitate optimization during downsampling and upsampling. For adversarial training, we extend the StyleGAN2 discriminator [58] to process video inputs.

#### A.2.2. Dataset

Our DC-AE-V is trained on a mixture of video and image datasets. The video datasets include subsets of Panda70m [59] and OpenVid1m [60]. The image datasets include ImageNet21k [61], Mapillary Vistas [62], DataComp [63], WiderFace [64], WiderPerson [65], TextCaps [66], and Unsplash [67].

Video Autoencoder	Config	Compress. Ratio	Panda70m				UCF101				ActivityNet				Kinetics600			
			PSNR ↑ SSIM ↑ LPIPS ↓	PSNR ↑ SSIM ↑ LPIPS ↓	PSNR ↑ SSIM ↑ LPIPS ↓	FVD ↓	PSNR ↑ SSIM ↑ LPIPS ↓	PSNR ↑ SSIM ↑ LPIPS ↓	FVD ↓	PSNR ↑ SSIM ↑ LPIPS ↓	PSNR ↑ SSIM ↑ LPIPS ↓	FVD ↓	PSNR ↑ SSIM ↑ LPIPS ↓	FVD ↓	PSNR ↑ SSIM ↑ LPIPS ↓	FVD ↓	PSNR ↑ SSIM ↑ LPIPS ↓	FVD ↓
VideoVAEPlus [21]	f8t4c16	48	36.88	0.968	0.009	35.79	0.959	0.016	2.11	35.68	0.955	0.016	0.96	36.73	0.960	0.014	0.93	
CogVideoX VAE [18]	f8t4c16	48	35.54	0.961	0.021	34.53	0.949	0.034	8.32	34.47	0.946	0.034	5.16	35.40	0.951	0.032	4.18	
HunyuanVideo VAE [19]	f8t4c16	48	35.46	0.960	0.015	34.40	0.950	0.024	3.80	34.41	0.943	0.024	3.16	35.40	0.950	0.022	2.59	
IV VAE [20]	f8t4c16	48	35.32	0.959	0.017	34.84	0.955	0.025	3.71	35.03	0.948	0.025	1.88	36.27	0.956	0.022	1.52	
Wan 2.1 VAE [5]	f8t4c16	48	34.15	0.952	0.017	33.81	0.943	0.024	3.71	33.82	0.938	0.025	1.76	35.04	0.946	0.022	1.49	
Wan 2.2 VAE [5]	f16t4c48	64	35.12	0.958	0.013	34.27	0.948	0.022	4.02	34.41	0.943	0.021	1.56	35.57	0.950	0.019	1.51	
StepVideo VAE [22]	f16t8c64	96	32.17	0.930	0.043	32.17	0.930	0.043	8.23	32.08	0.922	0.047	5.29	33.02	0.931	0.044	4.62	
Video DC-AE <sub>clipping &amp; blending</sub> [25]	f32t4c128	96	34.10	0.952	0.023	33.65	0.945	0.034	14.22	33.55	0.938	0.033	7.92	34.73	0.946	0.030	6.81	
Video DC-AE [25]	f32t4c128	96	31.73	0.915	0.040	31.52	0.914	0.047	26.30	31.34	0.901	0.049	17.52	32.39	0.915	0.044	15.43	
LTX Video VAE [24]	f32t8c128	192	32.41	0.928	0.039	31.12	0.910	0.059	70.92	31.29	0.900	0.058	45.51	32.26	0.911	0.056	42.30	
DC-AE-V	f32t4c256	48	39.56	0.979	0.008	37.14	0.967	0.018	1.95	37.29	0.965	0.016	0.89	38.12	0.969	0.015	0.82	
	f32t4c128	96	37.37	0.968	0.013	34.83	0.951	0.026	5.26	35.06	0.948	0.024	2.46	35.91	0.953	0.023	2.36	
	f32t4c64	192	35.03	0.953	0.019	32.71	0.931	0.035	12.15	33.02	0.927	0.034	5.64	33.87	0.934	0.032	5.70	
	f32t4c32	384	33.07	0.933	0.027	30.83	0.909	0.046	29.11	31.08	0.901	0.045	13.83	32.01	0.912	0.042	13.05	
	f64t4c128	384	32.79	0.932	0.030	30.60	0.907	0.048	29.35	30.86	0.898	0.048	13.63	31.73	0.909	0.046	13.60	

Table 5 | Additional Video Reconstruction Results.

### A.2.3. Evaluation

We evaluate all video autoencoders on  $80 \times 256 \times 256$  videos using PSNR, SSIM [68], LPIPS [69], and FVD [70]. The evaluation set includes 1,000 unseen videos from Panda70m [59], 3,783 test videos from UCF101 [71], 5,044 test videos from ActivityNet 1.3 [72], and the first 5,000 test videos from Kinetics600 [73].

For VideoVAEPlus [21], we use the ‘16z’ version. For CogVideoX VAE [18], we use the model from [THUDM/CogVideoX-2b](#). For HunyuanVideo VAE [19], we use the model from [hunyuanvideo-community](#). For IV VAE [20], we use the ‘ivvae\_z16\_dim96’ version. For Wan 2.1 VAE [5], we use the model from [Wan-AI/Wan2.1-T2V-14B-Diffusers](#), and for Wan 2.2 VAE [5], we use the model from [Wan-AI/Wan2.2-TI2V-5B](#). For StepVideo VAE [22], we use the ‘vae\_v2’ from [stepfun-ai/stepvideo-t2v](#). For Video DC-AE [25], we use the model from [hpcai-tech/Open-Sora-v2-Video-DC-AE](#). Finally, for LTX Video VAE [24], we use the model from [Lightricks/LTX-Video-0.9.7-dev](#). When an autoencoder cannot process 80-frame videos, we pad extra frames at the end and exclude them from the reconstructions when computing evaluation metrics.

### A.2.4. Additional Reconstruction Results

Table 5 presents the full reconstruction results. Our DC-AE-V consistently achieves superior accuracy and generalizes effectively to longer videos across a range of benchmarks.

Figure 10 presents additional reconstruction examples. Our DC-AE-V demonstrates superior reconstruction accuracy and generalization ability to longer videos, especially for small texts and human faces.

## A.3. Ablation Study on Video Embedding Space Alignment

Figure 11 presents additional ablation studies on video embedding space alignment. Aligning both the patch embedder and output head yields the best results, with the patch embedder alignment playing the most critical role in overall performance.

## A.4. Detailed Evaluation Results on VBench

Table 6 reports detailed metrics on VBench. DC-VideoGen-Wan-2.1-T2V-1.3B outperforms the base Wan-2.1-T2V-1.3B on 11 of the 16 metrics.

## A.5. Detailed Efficiency Benchmark Results

Table 7 presents detailed efficiency results measured on an NVIDIA H100 GPU.

## A.6. Detailed Training Hyperparameters of AE-Adapt-V

Table 8 lists the detailed hyperparameters for AE-Adapt-V.



Figure 10 | Additional Video Autoencoder Reconstruction Visualization. Under deep compression settings, causal video autoencoders suffer from low reconstruction quality. In contrast, non-causal video autoencoders achieve better reconstruction quality but generalize poorly to longer videos.

Text-to-Video Generation Results on VBench 720×1280								
Models	Temporal Flickering	Subject Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Background Consistency	Overall Consistency
Wan-2.1-T2V-1.3B	99.15	94.97	<b>98.36</b>	67.78	70.20	68.44	97.99	<b>25.97</b>
DC-VideoGen-Wan-2.1-T2V-1.3B	<b>99.18</b>	<b>96.58</b>	98.34	<b>72.78</b>	<b>72.00</b>	<b>68.72</b>	<b>98.00</b>	25.41
Models	Object Class	Multiple Objects	Human Action	Color	Spatial Relationship	Scene	Appearance Style	Temporal Style
Wan-2.1-T2V-1.3B	<b>89.11</b>	72.07	94.05	81.54	65.90	44.83	<b>21.61</b>	<b>23.22</b>
DC-VideoGen-Wan-2.1-T2V-1.3B	88.73	<b>75.98</b>	<b>94.64</b>	<b>89.16</b>	<b>78.20</b>	<b>44.86</b>	21.20	22.97

Table 6 | Detailed Results on VBench.

(a) Latency (minutes per video)					(b) Latency (minutes per video)				
Models	Resolution				Number of Frames	80	160	320	640
	480×832	720×1280	1080×1920	2160×3840					
Wan-2.1-1.3B [5]	1.49	5.76	25.46	375.12					
DC-VideoGen-Wan-2.1-1.3B	0.24	0.70	2.27	25.41					
Speedup	6.2×	8.2×	11.2 ×	14.8×					

Table 7 | Detailed Efficiency Benchmark Results.

#### A.7. Qualitative Comparison with the Pre-trained Models

Figure 12 and Figure 13 provide a qualitative comparison between DC-VideoGen and the base models. We observe that DC-VideoGen-Wan2.1-I2V-14B and DC-VideoGen-Wan2.1-T2V-14B retain the generation quality

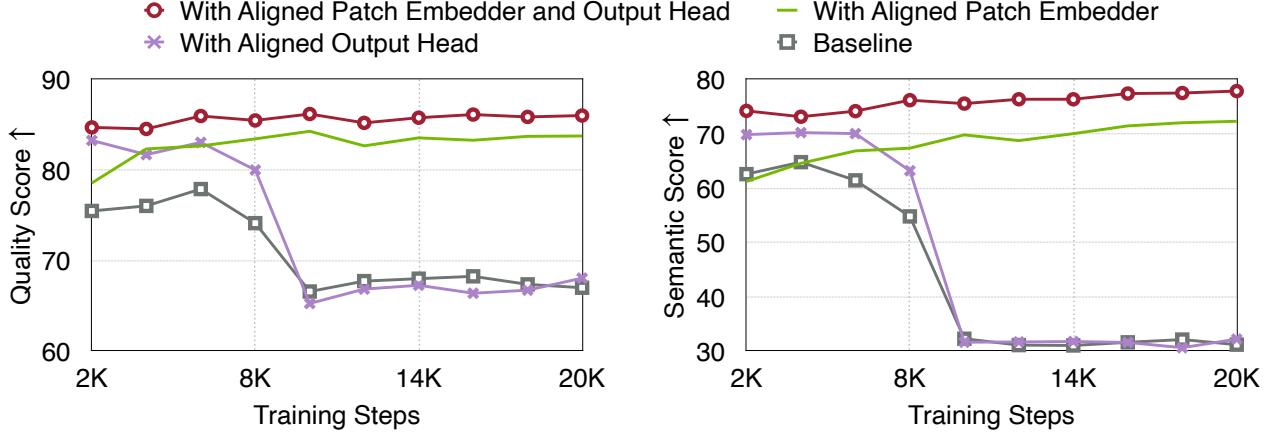


Figure 11 | Ablation Study on Video Embedding Space Alignment.

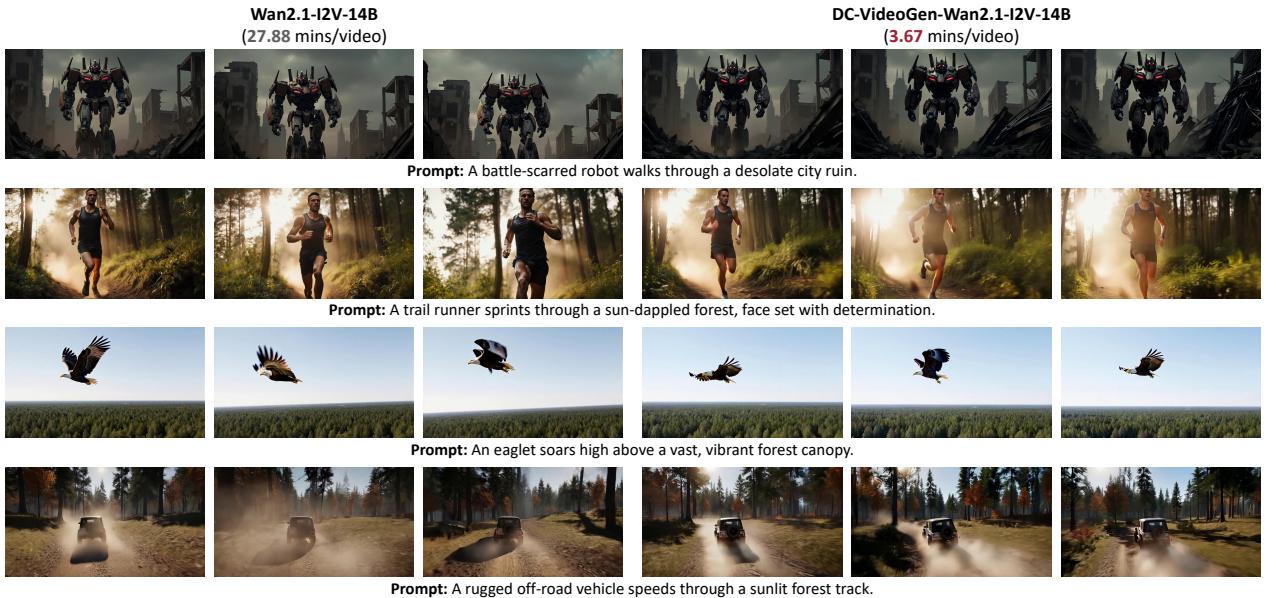


Figure 12 | Visual Comparison of DC-VideoGen-Wan2.1-I2V-14B and the Base Model Wan2.1-I2V-14B.

of Wan2.1-I2V-14B and Wan2.1-T2V-14B while reducing the latency by around 87%.

### A.8. Limitation and Future Work

**Limitations.** As a post-training framework, DC-VideoGen accelerates video diffusion models through lightweight fine-tuning, effectively leveraging the rich knowledge encoded in the pre-trained model. Consequently, its performance is strongly dependent on the quality of the pre-trained model.

**Future Work.** DC-VideoGen substantially reduces the training and inference costs of video diffusion models, especially when scaling to higher resolutions. For the next step, we plan to extend our framework for long video generation, leveraging techniques from [74, 75].

Training Stage	Hyperparameter	Value
Patch Embedder Alignment	learning rate	1e-4
	warmup steps	0
	batch size	4
	training steps	20k
	optimizer	AdamW, betas=[0.9, 0.999]
Output Head Alignment	learning rate	1e-4
	warmup steps	0
	batch size	32
	training steps	4k (Wan-2.1-1.3B) / 3k (Wan-2.1-14B)
	optimizer	AdamW, betas=[0.9, 0.999]
End-to-End Fine-Tuning	learning rate	5e-5
	warmup steps	1k
	training steps	20k (Wan-2.1-1.3B) / 6k (Wan-2.1-14B)
	batch size	32
	optimizer	AdamW, betas=[0.9, 0.999]
	weight decay	1e-3
Resolution Increasing	LoRA (rank, alpha)	(256, 512)
	480px→720px, training steps	1000
	720px→1080px, training steps	500
	1080px→2160px, training steps	200

Table 8 | Training Hyperparameters of AE-Adapt-V.

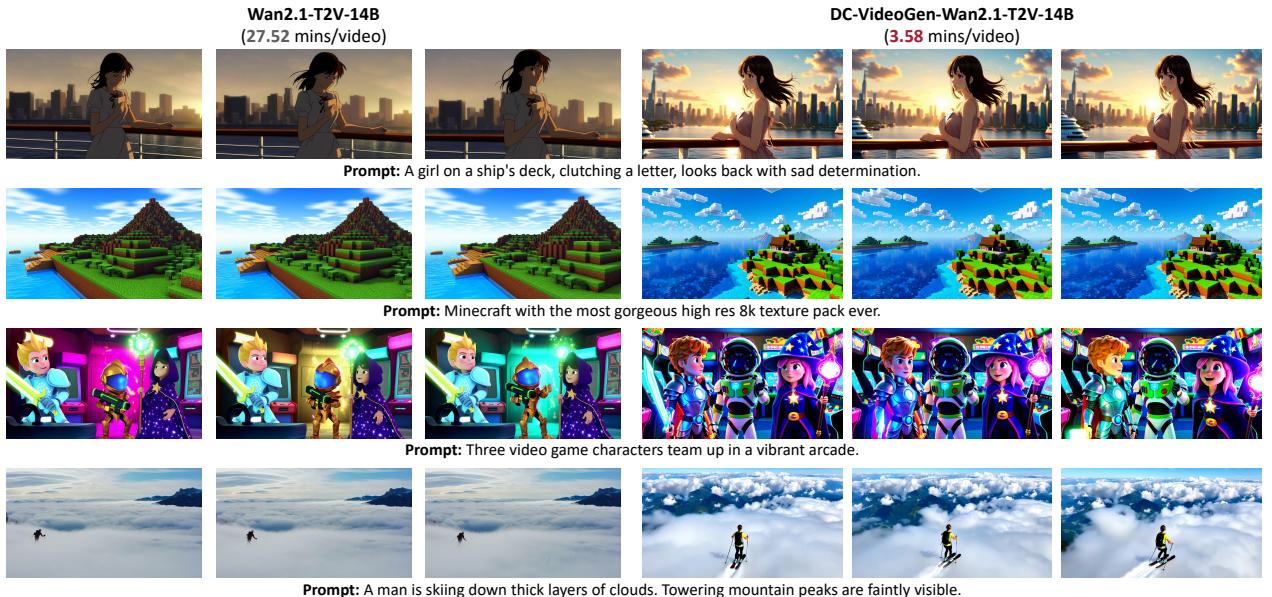


Figure 13 | Visual Comparison of DC-VideoGen-Wan2.1-T2V-14B and the Base Model Wan2.1-T2V-14B.