SF-V: Single Forward Video Generation Model

Zhixing Zhang[1,2*] **Yanyu Li**[1] **Yushu Wu**[1] **Yanwu Xu**[1] **Anil Kag**[1]
Ivan Skorokhodov[1] **Willi Menapace**[1] **Aliaksandr Siarohin**[1] **Junli Cao**[1]
Dimitris Metaxas[2] **Sergey Tulyakov**[1] **Jian Ren**[1†]
[1]Snap Inc. [2] Rutgers University

Project Page: https://snap-research.github.io/SF-V

Figure 1: Example generation results from our *single*-step image-to-video model. Our model can generate high-quality and motion consistent videos by only performing the sampling *once* during inference. Please refer to our webpage for whole video sequences.

Abstract

Diffusion-based video generation models have demonstrated remarkable success in obtaining high-fidelity videos through the iterative denoising process. However, these models require multiple denoising steps during sampling, resulting in high computational costs. In this work, we propose a novel approach to obtain *single*-step video generation models by leveraging adversarial training to fine-tune pre-trained video diffusion models. We show that, through the adversarial training, the multi-steps video diffusion model, *i.e.*, Stable Video Diffusion (SVD), can be trained to perform *single* forward pass to synthesize high-quality videos, capturing both temporal and spatial dependencies in the video data. Extensive experiments demonstrate that our method achieves competitive generation quality of synthesized videos with significantly reduced computational overhead for the denoising process (*i.e.*, around $23\times$ speedup compared with SVD and $6\times$ speedup compared with existing works, with even better generation quality), paving the way for real-time video synthesis and editing.

[*]Work done during an internship at Snap Inc.
[†]Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3290

1 Introduction

Video generation is experiencing unprecedented advancements by leveraging large-scale denoising diffusion probabilistic models [1, 2] to create photo-realistic frames with natural and consistent motion [3, 4], revolutionizing various fields, such as entertainment and digital content creation [5, 6].

Early efforts on image generation show that diffusion models have the significant capabilities when scaled-up to generate diverse and high-fidelity content [1, 2]. Additionally, these models benefit from a stable training and convergence process, demonstrating a considerable improvement over their predecessors, *i.e.*, generative adversarial networks (GANs) [7]. Therefore, many studies on video generation are built upon the diffusion models. Some of them utilize the pre-trained image diffusion models for video synthesis through introducing temporal layers to generate high-quality video clips [8, 9, 10, 11]. Inspired by this design paradigm, numerous video generation applications have emerged, such as animating a given image with optional motion priors [12, 13, 14, 15], generating videos from natural language descriptions [16, 17, 5], and even synthesizing cinematic and minutes-long temporal-consistent videos [18, 4].

Despite the impressive generative performance, video diffusion models suffer from tremendous computational costs, hindering their widespread and efficient deployment. The iterative nature of the sampling process makes video diffusion models significantly slower than other generative models (*e.g.*, GANs [19, 20]). For instance, in our benchmark, it only takes 0.3 seconds to perform a single denoising step using the UNet from the Stable Video Diffusion (SVD) [13] model to generate 14 frames on one NVIDIA A100 GPU, while consuming 10.79 seconds to run the UNet with the conventional 25-step sampling.

The significant overhead introduced by iterative sampling highlights the necessity to generate videos in fewer steps while maintaining the quality of multi-step sampling. Recent works [21, 22, 23] extend consistency training [24] to video diffusion models, offering two main benefits: reduced total runtime by performing fewer sampling steps and the preservation of the pre-trained ordinary differential equation (ODE) trajectory, allowing high-quality video generation with fewer sampling steps (*e.g.*, 8 steps). Nevertheless, these approaches still struggle to achieve *single*-step high-quality video generation.

On the other hand, distilling image diffusion models into one step via adversarial training have shown promising progress [25, 26, 27, 28, 29]. However, scaling up such approaches for video diffusion model training to achieve single-step generation has not been well studied. In this work, we leverage adversarial training to obtain an image-to-vide o generation model that requires only *single*-step generation, with the contributions summarized as follows:

- We build the framework to fine-tune the pre-trained state-of-the-art video diffusion model (*i.e.*, SVD) to be able to generate videos in *single* forward pass, greatly reducing the runtime burden of video diffusion model. The training is conducted through adversarial training on the latent space.
- To improve the generation quality (*e.g.*, higher image quality and more consistent motion), we introduce the discriminator with spatial-temporal heads, preventing the generated videos from collapsing to the conditional image.
- We are the first to achieve one-step generation for video diffusion models. Our one-step model demonstrates superiority in FVD [30] and visual quality. Specifically, for the denoising process, our model achieves around $23\times$ speedup compared with SVD and $6\times$ speedup compared with exiting works, with even better generation quality.

2 Related Work

Video Generation has been a long studied problem, aiming for high-quality image generation and consistent motion synthesis. Early efforts in this domain utilize adversarial training [31, 32]. Though extensively investigated, the trained models still suffer from low resolution, limited generated sequences, and inconsistent motion. Recent studies leverage denoising diffusion probabilistic models [1, 33, 34] to scale the video generators up to billions of model parameters, achieving high-fidelity generation sequences [35, 36, 37, 38, 39, 5, 4, 3, 18]. Nonetheless, the tremendous computation cost of video diffusion models hinders their wide deployment. It takes tens of seconds to generate a

single video batch even for high-tier server GPUs. Consequently, the reduction of denoising steps [21, 40, 22] is pivotal to efficient video generation, which linearly scales down the total runtime.

Step Distillation of Diffusion Models. Initially developed upon image diffusion models, progressive distillation [41, 42] aims to distill a less-step student mimicking the full-step counterpart. Specifically, at each step, the student learns to predict a teacher location in the ODE flow, resulting in fewer required denoising steps during inference time. Latent Consistency Models (LCM) [24, 43, 44, 45, 46, 47, 48] instead proposes to refine the prediction objective into clean data, and achieves high-fidelity generation with fewer $(2 \sim 4)$ steps. Rectified flow [49, 50] progressively straights the ODE flow where each denoising step becomes a substitution of a long trajectory. UFOGen [25], ADD [27], and its latent-space successor LADD [28] further incorporate adversarial loss to distill teacher signal into the few-step student, enabling one-step generation with reasonable quality, and outperforming the teacher model with about 4 steps. DMD [26] proposes to combine a distribution matching objective and a regression loss to distill a one-step generator. The recent SDXL-Lightning [29] combines progressive distillation with adversarial loss to mitigate the blurry generation issue and ease the convergence of multi-step settings. In addition, SDXL-Lightning refines the design of the discriminator and proposes two adversarial loss objectives to balance sample quality and mode convergence.

When it comes to video models, VideoLCM [40] and AnimateLCM [21] adopt consistency distillation to enable 4-step generation with comparable quality to the full-step pre-trained video diffusion model. However, in the one-step setting, there are still considerable performance gaps observed for the visual quality. Animate-Diff Lightning [22] incorporates adversarial distillation to further reduce warps and blurs in the 1-2 step setting, despite that the model still underperforms full-step baselines.

3 Method

Our goal is to generate high-fidelity and temporally consistent videos in as few sampling steps as possible (*i.e.*, 1 step). The adversarial objective has been proven effective in reducing the number of sampling steps required by diffusion models in image space [27, 28, 25, 51]. However, limited efforts have been conducted on scaling up the effective adversarial training to reduce the number of sampling steps for video diffusion models. In the following, we introduce the framework of latent adversarial training to obtain efficient video diffusion model by running sampling in *single* step. In this framework, we initialize the generator and part of the discriminator with the weights of a pre-trained video diffusion model. Moreover, we introduce a structure with separate spatial and temporal discriminator heads to enhance frame quality and motion consistency.

3.1 Preliminaries of Stable Video Diffusion

Our method is built upon the Stable Video Diffusion (SVD) [13], which is an implementation of the EDM-framework [33] for conditional video generation, where the diffusion process is conducted in latent space. We choose the *publicly released* image-to-video generation pipeline of SVD due to its superior performance in generating high-quality and motion-consistent videos.

Training Diffusion Models with EDM. To facilitate the presentation, let $p_{data}(x_0)$ denote the data distribution and $p(x;\sigma)$ represent the distribution obtained by adding σ^2-variance Gaussian noise to the data. For sufficiently large $\sigma_{\max}$, $p(x;\sigma_{\max}) \approx \mathcal{N}(0, \sigma^2_{\max})$. Starting from high variance Gaussian noise $x_M \sim \mathcal{N}(0, \sigma^2_{\max})$, the diffusion models sequentially denoise towards $\sigma_0 = 0$ through the numerical simulation of the *Probability Flow* ODE [52]. The denoiser, D_θ, attempts to predict the clean x_0 and is trained via denoising score matching:

$$\mathbb{E}_{x_0 \sim p_{data}(x_0), (\sigma, \mathbf{n}) \sim p(\sigma, \mathbf{n})} \left[\lambda_\sigma \| D_\theta(x_0 + \mathbf{n}; \sigma) - x_0 \|_2^2 \right], \tag{1}$$

where $p(\sigma, \mathbf{n}) = p(\sigma)\mathcal{N}(\mathbf{n}; 0, \sigma^2)$, $p(\sigma)$ is a distribution over noise levels σ, and $\lambda_\sigma : \mathbb{R}^+ \to \mathbb{R}^+$ is a weighting function.

EDM [33] parameterizes the denoiser D_θ as:

$$D_\theta(x;\sigma) = c_{skip}(\sigma)x + c_{out}(\sigma)F_\theta(c_{in}(\sigma)x; c_{noise}(\sigma)), \tag{2}$$

where F_θ is the network to be trained. The preconditioning functions are set as $c_{skip}(\sigma) = (\sigma^2+1)^{-1}$, $c_{out}(\sigma) = \frac{-\sigma}{\sqrt{\sigma^2+1}}$, $c_{in}(\sigma) = \frac{1}{\sqrt{\sigma^2+1}}$, and $c_{noise}(\sigma) = 0.25 \log \sigma$.

https://doi.org/10.52202/079017-3290

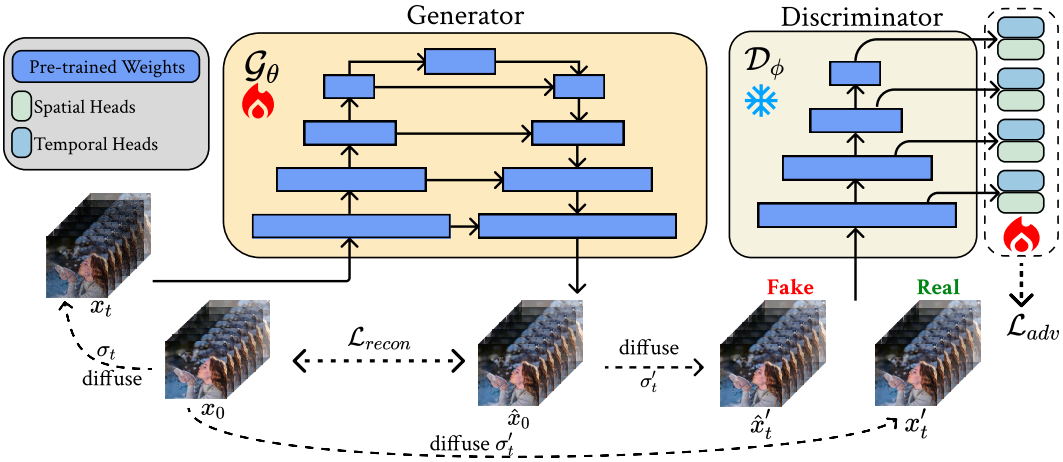

Figure 2: **Training Pipeline.** We initialize our generator and discriminator using the weights of a pre-trained image-to-video diffusion model. The discriminator utilizes the encoder part of the UNet as its backbone, which remains *frozen* during training. We add a spatial discriminator head and a temporal discriminator head after each downsampling block of the discriminator backbone and only update the parameters of these heads during training. Given a video latent x_0, we first add noise σ_t through a forward diffusion process to obtain x_t. The generator then predicts $\hat{x}_0$ given x_t. We calculate the reconstruction loss $\mathcal{L}_{recon}$ between x_0 and $\hat{x}_0$. Additionally, we add noise level σ'_t to both x_0 and $\hat{x}_0$ to obtain real and fake samples, x'_t and $\hat{x}'_t$. The adversarial loss $\mathcal{L}_{adv}$ is then calculated using these real and fake sample pairs.

Stable Video Diffusion. The training of video model asks for a dataset of videos, each consisting of N frames with height H and width W. Given a video $\mathbf{V}_0 = \{\mathbf{I}_0^i\}_{i=0}^N$, where $\mathbf{I}_0^i \in \mathbb{R}^{3\times H\times W}$, SVD [13] maps each frame separately to latent space using a frame encoder, E. The encoded frames are represented as $x_0 = \{E(\mathbf{I}_0^i)\}_{i=0}^N$, resulting in $x_0 \in \mathbb{R}^{N\times 4\times \tilde{H}\times \tilde{W}}$. Here, $x_0 \sim p_{data}(x_0)$ is a sequence of N latent frames with 4 channels, height $\tilde{H}$, and width $\tilde{W}$.

SVD inflates a text-to-image diffusion model to a text-to-video diffusion model [10]. The text conditioning is replaced with image conditioning to create an image-to-video diffusion model. Consequently, the parameterized denoiser D_θ in Eq. (2) is modified as follows:

$$D_\theta(x;\sigma,\mathbf{c}) = c_{skip}(\sigma)x + c_{out}(\sigma)F_\theta(c_{in}(\sigma)x; c_{noise}(\sigma),\mathbf{c}), \tag{3}$$

where $\mathbf{c}$ is the image condition $\mathbf{I}_0^0$, *i.e.*, the first frame of the video.

At sampling time, D_θ is leveraged to restore x_{t-1} from x_t using the following relation [33]:

$$d_t = (x_t - D_\theta(x_t;\sigma_t,\mathbf{c}))/\sigma_t; \;\; x_{t-1} = x_t + (\sigma_{t-1}-\sigma_t)\cdot d_t, \tag{4}$$

where σ_t is obtained with

$$\sigma_t = (\sigma_{\min}^{1/\rho} + \frac{t}{T-1}(\sigma_{\max}^{1/\rho} - \sigma_{\min}^{1/\rho}))^\rho, \tag{5}$$

where T is the total number of denoising steps and ρ is a hyper-parameter controlling the emphasis level to low noise levels.

3.2 Latent Adversarial Training for Video Diffusion Model

Design of Networks. Diffusion-GAN hybrid models are designed for training with large denoising step sizes [25, 27, 28, 51]. Our training procedure, illustrated in Fig. 2, involves two networks: a generator $\mathcal{G}_\theta$ and a discriminator $\mathcal{D}_\phi$. The generator is initialized from a pre-trained UNet diffusion model with weights θ (*i.e.*, the UNet from SVD). The discriminator is *partially* initialized from a pre-trained UNet diffusion model. Namely, the backbone of the discriminator shares the same architecture and weights as the pre-trained UNet encoder, and the weights of this backbone are kept frozen during training. Additionally, we *augment* the discriminator by adding a spatial discriminator head and

https://doi.org/10.52202/079017-3290

a temporal discriminator head after each backbone block. Therefore, in total, the discriminator comprises four spatial discriminator heads and four temporal discriminator heads. Only the parameters in these heads are trained during the discriminator training steps. The detailed architecture of these heads will be further discussed in Sec. 3.3.

Latent Adversarial Training. We use a pair of generated samples $\hat{x}_0$ and real samples x_0 to conduct the adversarial training. Specifically, during training, the generator $\mathcal{G}_\theta$ produces *generated* samples $\hat{x}_0(x_t; \sigma_t, \mathbf{c})$ from noisy data x_t. The noisy data points are derived from a dataset of *real* latents x_0 via a forward diffusion process $x_t = x_0 + \sigma_t \epsilon$. We sample σ_t uniformly from the set $\{\sigma_1, \cdots, \sigma_{T_g - 1}\}$, obtained by setting T to T_g and $t \in \{1, 2, \cdots, T_g - 1\}$ in Eq. (5). In practice, we set $T_g = 4$. The generated sample $\hat{x}_0$ is given by:

$$\hat{x}_0(x_t; \sigma_t, \mathbf{c}) = c_{skip}(\sigma_t) x_t + c_{out}(\sigma_t) \mathcal{G}_\theta(c_{in}(\sigma_t) x_t; c_{noise}(\sigma_t), \mathbf{c}). \tag{6}$$

To train the discriminator, we forward the generated samples $\hat{x}_0$ and real samples x_0 into it, aiming to let the discriminator distinguish between them. However, for a more stabilized training, inspired by exiting works [28], we add noise to the samples before passing them to the discriminator, since the backbone of the discriminator is initialized from a pre-trained UNet with weights frozen during training. Namely, we sample σ'_t from the set $\{\sigma'_1, \cdots, \sigma'_{T_d - 1}\}$, obtained by setting T to T_d and $t \in \{1, 2, \cdots, T_d - 1\}$ in Eq. (5), according to a discretized lognormal distribution defined as:

$$p(\sigma'_t) \propto erf\left(\frac{\log(\sigma'_t - P_{mean})}{\sqrt{2} P_{std}}\right) - erf\left(\frac{\log(\sigma'_{t-1} - P_{mean})}{\sqrt{2} P_{std}}\right), \tag{7}$$

where P_{mean} and P_{std} control the noise level added to the samples before passing them to the discriminator. A visualization of how different P_{mean} and P_{std} affect the probability of σ' sampled is illustrated in Fig. 6. In practice, we set $T_d = 1,000$. We diffuse the real and generated samples through the forward process to obtain $\hat{x}'_t = \hat{x}_0 + \sigma'_t \epsilon$ and $x'_t = x_0 + \sigma'_t \epsilon$, respectively.

Following literature [27, 53, 54], we use the hinge loss [55] as the adversarial objective function for improved performance. The adversarial optimization for the generator $\mathcal{L}^{\mathcal{G}}_{adv}(\hat{x}_0, \phi)$ is defined as:

$$\mathcal{L}^{\mathcal{G}}_{adv} = \mathbb{E}_{\sigma, \sigma', x_0}[\mathcal{D}_\phi\left(c_{in}(\sigma'_t)\hat{x}'_t\right)], \tag{8}$$

Furthermore, we notice that a reconstruction objective, $\mathcal{L}_{recon}$, between x_0 and $\hat{x}_0$ can significantly improve the stability of the training process. We use Pseudo-Huber metric [56, 43] for reconstruction loss, as:

$$\mathcal{L}_{recon}(\hat{x}_0, x_0) = \sqrt{\|\hat{x}_0 - x_0\|_2^2 + c^2} - c, \tag{9}$$

where $c > 0$ is an adjustable constant. Thus, the overall objective for training the generator is as follows with λ balances two losses:

$$\mathcal{L}^{\mathcal{G}} = \mathcal{L}^{\mathcal{G}}_{adv} + \lambda \mathcal{L}_{recon}(\hat{x}_0, x_0). \tag{10}$$

Other other hand, the discriminator is trained to minimize:

$$\mathcal{L}^{\mathcal{D}}_{adv} = \mathbb{E}_{\sigma', x_0}[\max(0, 1 + \mathcal{D}_\phi\left(c_{in}(\sigma'_t)x'_t\right)) + \gamma R1] + \mathbb{E}_{\sigma, \sigma', x_0}[\max(0, 1 - \mathcal{D}_\phi\left(c_{in}(\sigma'_t)\hat{x}'_t\right)))], \tag{11}$$

where $R1$ denotes the R1 gradient penalty [57, 27]. Here, we omit other conditional input for $\mathcal{D}_\phi$, such as $c_{noise}(\sigma')$ and image conditioning $\mathbf{c}$, for simplicity.

Discussion. Our latent adversarial training framework is largely inspired by LADD [28]. Similar to LADD, we set $T_g = 4$ in practice and utilize a pre-trained diffusion model as part of the discriminator. However, our approach has several key differences compared with LADD [28]. *First*, we extend the image latent adversarial distillation framework to the video domain by incorporating spatial and temporal heads to achieve one-step generation for video diffusion models. The specifics of the spatial and temporal heads are discussed in Sec. 3.3. *Second*, based on the EDM-framework [33], we observe that sampling t' using a discretized lognormal distribution provides more stable adversarial training compared to the logit-normal distribution used in LADD [28]. *Finally*, unlike LADD [28], we utilize real video data instead of synthetic data for training and incorporate a reconstruction objective (*i.e.*, Eq. (9)) to ensure more stable training.

https://doi.org/10.52202/079017-3290

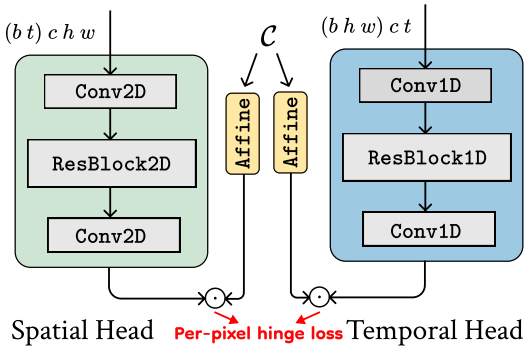

Figure 3: **Spatial & Temporal Discriminator Heads.** Our discriminator heads take in intermediate features of the UNet encoder. Follow existing arts [54, 53], we use image conditioning and frame index as the projected condition $\mathbf{c}$. **Left:** For spatial discriminator heads, the input features are reshaped to merge the temporal axis and the batch axis, such that each frame is considered as an independent sample. **Right:** For temporal discriminator heads, we merge spatial dimensions to batch axis.

Table 1: **Comparison Results.** We compare our method against SVD [13], AnimateLCM [21], UFOGen [25], and LADD [28] using different numbers of sampling steps. AnimateLCM* indicates the usage of the officially provided 25-frame model, with only the first 14 frames considered for FVD calculation. † indicates our implementations. We also report the latency of the denoising process for each setting, measured on a single NVIDIA A100 GPU.

Name	FVD↓	Steps	Latency (s)
SVD [13]	153.4	25	10.79
	194.4	16	6.89
	488.6	8	3.44
	1687.0	4	1.72
AnimateLCM* [21]	321.1	8	3.25
	403.2	4	1.62
	521.9	2	0.82
AnimateLCM [21]	281.0	8	1.85
	801.4	4	0.92
	1158.4	2	0.46
UFOGen† [25]	1917.2	1	0.30
LADD† [28]	1893.8	1	0.30
Ours	180.9	1	0.30

3.3 Spatial Temporal Heads

To train the discriminator for better understanding of the spatial information and temporal correlation, we employ separate spatial and temporal discriminator heads for adversarial training [31, 32]. The backbone of the discriminator is the encoder from the pre-trained diffusion model (*i.e.*, UNet), which consists of four spatial-temporal blocks sequentially [10]. The first three blocks downsample the spatial resolution by a factor of 2, and the last block maintains the spatial resolution. We extract the output features from each spatial-temporal block and utilize a spatial head and a temporal head to determine whether the sample is real or fake. The discriminator can be conditioned on additional information via projection [58] to enhance performance. In our setting, we use the image condition $\mathbf{c}$ and σ' as the projected condition $\mathcal{C}$.

Spatial Head. For an input feature of shape $b \times t \times c \times h \times w$, the spatial discriminator first reshapes it to $(bt) \times c \times h \times w$. This way, each frame feature in a video is processed separately. The architecture for our proposed spatial head is illustrated in the left part of Fig. 3.

Temporal Head. Even though the features obtained from the discriminator backbone contain spatial-temporal information, we observe that using only spatial discriminator heads causes the generator to produce frames that are all identical to the image condition. To achieve better temporal performance (*e.g.*, more vivid motion), we propose to add a temporal discriminator head parallel to the spatial discriminator head. The input features are reshaped to $(bhw) \times c \times t$ instead. The architecture for our temporal head is illustrated in the right part of Fig. 3.

4 Experiment

Implementation Details. We apply Stable Video Diffusion [13] as the base model across our experiments. All the experiments are conducted on an internal video dataset with around one million videos. We fix the resolution of the training videos as 768×448 with the FPS as 7. The training is conducted for 50K iterations on 8 NVIDIA A100 GPUs, using the SM3 optimizer [59] with a learning rate of $1e-5$ for the generator (*i.e.*, UNet) and $1e-4$ for the discriminator. We set the momentum and β for both optimizers as 0.5 and 0.999, respectively. The total batch size is set as 32 using a 4

Figure 4: **Video Generation on Single Conditioning Images from Various Domains.** We employ our method on various images generated by SDXL [60] to synthesized videos. The videos contain 14-frame at a resolution of 1024×576 with 7 FPS. The results demonstrate that our model can generate high-quality motion-consistent videos of various objects across different domains. Please refer to our webpage for whole video sequences.

steps gradient accumulation. We set the EMA rate as 0.95. We set $P_{mean} = -1, P_{std} = -1$, and $\lambda = 0.1$ if not otherwise noted. At inference time, we sample videos at resolution of 1024×576.

4.1 Qualitative Visualization

To comprehensively evaluate the capabilities of our method, we use SDXL [60] (with refiner) to generate images of different scenes at the resolution of 1024×1024. We then perform center crop on the generated images to get resolution as 1024×576, which serves as the condition of our approach to synthesize videos of 14 frames at 7 FPS. As shown in Fig. 4, our method can successfully generate videos of high-quality frames and consistent object movements with *only* 1 step during inference.

4.2 Comparisons Results

Quantitative Comparisons. We present a comprehensive evaluation of our method compared to the existing state-of-the-art approach, AnimateLCM [21], UFOGen [25], LADD [28], and SVD [13]. To

https://doi.org/10.52202/079017-3290

Figure 5: **Comparison between SVD [13], AnimateLCM [21], LADD [28], UFOGen [25], and Our Approach.** We provide the synthesized videos (sampled frames) under various settings for different approaches. We use SVD to generate videos under 25, 16, and 8 sampling steps, AnimateLCM to synthesize videos under 4 sampling steps, LADD and UFOGen to generate videos under 1 sampling step. AnimateLCM, LADD and UFOGen generates blurry frames with few-steps and single-step sampling. Our approach can accelerate the sampling speed by $22.9\times$ compared with SVD while maintaining similar frame quality and motion consistency.

https://doi.org/10.52202/079017-3290

conduct a fair comparison on the SVD model, we train the AnimateLCM, UFOGen, and LADD on SVD using our video dataset. We follow the released code and instructions provided by AnimateLCM authors. Additionally, we include the officially released AnimateLCM-xt1.1 [21] by evaluating the first 14 generated frames and denote the approach as AnimateLCM*. We try our best to implement LADD [28] and UFOGen [25] and denote respectively as LADD†, and UFOGen†. Note that simply re-using the discriminator from LADD [28] and UFOGen [25] leads to *out-of-memory issue*, since the computation in the video model is much larger than the image model. Here we replace the discriminator from LADD [28] and UFOGen [25] with the one proposed in our work.

We follow exiting works [61] by using Fréchet Video Distance (FVD) [30] as the comparison metric. Specifically, we use the first frame from the UCF-101 dataset [62] as the conditioning input and generate 14-frame videos at a resolution of 1024×576 at 7 FPS for all methods. The generation results are then resized back to 320×240 for FVD calculation. Our method is compared against SVD [13] and AnimateLCM [21], each using a different number of sampling steps. Furthermore, to better demonstrate the effectiveness of our method, we measure the generation latency of each method, which is calculated on running the diffusion model (*i.e.*, UNet). Note that only for SVD [13], classifier-free guidance [63] is used, leading to higher computational cost.

As shown in Tab. 1, our method achieves comparable results to the base model using 16 discrete sampling steps, resulting in approximately a $23\times$ speedup. Our method also outperforms the 8-steps sampling results for AnimateLCM and AnimateLCM*, indicating a speedup of more than $6\times$. For *single-step* evaluation, our method performs much better than existing step-distillation methods [25, 28] built upon image-based-diffusion models.

Qualitative Comparisons. We further provide qualitative comparisons across different approaches by using publicly available web images. Fig. 5 presents generation results from SVD [13] with 25, 16, and 8 sampling steps, AnimateLCM [21] with 4 sampling steps, UFOGen [25], LADD [28], and our method with 1 sampling step. As can be seen, our method achieves results comparable to the sampling results of SVD using 16 or 25 denoising steps. We notice significant artifacts for videos synthesized by SVD when using 8 denoising steps. Compared to AnimateLCM [21],UFOGen [25], and LADD [28], our method produces frames of higher quality and better temporal consistency, with fewer or same denoising steps, demonstrating the effectiveness of our proposed approach.

4.3 Ablation Analysis

Effect of Discriminator Heads. We explore the effect of our proposed spatial and temporal heads by measuring the FVD on the UCF-101 dataset. We conduct latent adversarial training with three different discriminator settings to analyze the impact of our spatial and temporal discriminators. As shown in Tab. 2, training with only spatial heads (denoted as *SP*) or only temporal heads (denoted as *TE*) results in significantly worse performance than using all of them (denoted as *SP+TE*).

Nevertheless, since our discriminator backbone shares the same architecture as the spatial-temporal generator, the receptive field of each pixel on the feature maps provided by the backbone can cover a region both spatially and temporally. Additionally, we embed the frame index as an additional projected condition. Consequently, even when using only spatial heads or only temporal heads, the generated videos still exhibit reasonable frame quality and temporal coherence.

Effect of Noise Distribution for Discriminator. As shown in Fig. 6, following Eq. (5), P_{mean} and P_{std} control the distribution of σ'_t, which is the noise level added to x_0 or $\hat{x}_0$ before passing to the discriminator as real and fake samples, respectively. We explore the effect of different noise distributions on model performance by calculating FVD on the UCF-101 dataset.

When the sampled σ'_t is concentrated on small values, *e.g.*, $P_{mean} = -2$ and $P_{std} = -1$ in our case, we notice that the discriminator can quickly learn to distinguish real samples from fake ones. This leads to a significant drop in performance, as shown in Tab. 3 and Fig. 7.

On the other hand, when the noise level becomes too high, *e.g.*, $P_{mean} = 1$ and $P_{std} = 1$, the discriminator input, which is $c_{\text{in}}(\sigma'_t)\hat{x}'_t = \frac{\hat{x}_0 + \sigma'_t \epsilon}{\sqrt{\sigma_t'^2 + 1}}$, results in small adversarial gradients for the generator. This causes increased artifacts in the generated videos, as shown in Fig. 7 and Tab. 3.

https://doi.org/10.52202/079017-3290

Table 2: **Analysis of discriminator.** We measure FVD for models with different discriminator configurations. "SP" indicates that spatial heads and "TE" indicates temporal heads.

	SP+TE	SP	TE
FVD	**180.9**	514.7	539.2

Table 3: **FVD *vs.* σ' distributions.**

P_{mean}	P_{std}	FVD
-2.0	-1.0	3370.4
-1.0	-1.0	**180.9**
0.0	1.0	416.7
1.0	1.0	632.9

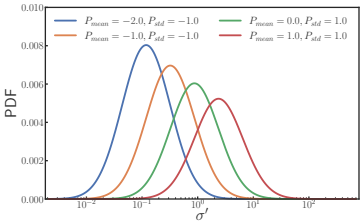

Figure 6: **PDF of σ'.**

Figure 7: **Analysis of σ' Distributions.** We investigate the impact of changing the distribution of σ' by adjusting P_{mean} and P_{std}. The results are shown with the same image conditioning. The first row and the second row display the first and last frames generated, respectively.

5 Discussion and Conclusion

In this work, we leverage adversarial training to reduce the denoising steps of the video diffusion model and thus improve its generation speed. We further enhance the discriminator by introducing spatial-temporal heads, resulting in better video quality and motion diversity. We are the first to achieve *1-step* generation for video diffusion models while preserving comparable visual quality and FVD scores, democratizing efficient video generation to a broader audience by delivering more than $20\times$ speedup for the denosing process.

Figure 8: **Limitations.** We show that, for some conditional images, our model tends to generate a few unsatisfactory frames when complex motion might be required (*Second Row*). Similar artifacts can also be observed in frames generated from SVD by sampling at 25-steps (*First Row*).

Limitations. We observe that when the given conditioning image indicates complex motion, *e.g.*running, our model tends to generate unsatisfactory results, *e.g.*blurry frames, as shown in Fig. 8. Such artifacts are introduced by the original SVD model, as can be observed in Fig. 8. We believe a better text-to-video model can solve such issue.

This work successfully achieves *single* sampling step for video diffusion models. However, under such setting, the temporal VAE decoder and the encoder for image conditioning take a considerable portion of the overall runtime. We leave the acceleration of these models as future work.

https://doi.org/10.52202/079017-3290

Acknowledgments and Disclosure of Funding

This research has been partially funded by grants to D. Metaxas from NSF: 2310966, 2235405, 2212301, 2003874, 1951890, AFOSR 23RT0630, and NIH 2R01HL127661.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[2] 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.

[3] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023.

[4] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Joshua V. Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. Videopoet: A large language model for zero-shot video generation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[5] Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stoliar, Elisa Ricci, Jian Ren, et al. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7038–7048, 2024.

[6] Zhixing Zhang, Bichen Wu, Xiaoyan Wang, Yaqiao Luo, Luxin Zhang, Yinan Zhao, Peter Vajda, Dimitris Metaxas, and Licheng Yu. Avid: Any-length video inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7162–7172, 2024.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[8] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[9] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[10] 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.

[11] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[12] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. *arXiv preprint arXiv:2311.16498*, 2023.

[13] 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.

 https://doi.org/10.52202/079017-3290

[14] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023.

[15] Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Animateanything: Fine-grained open domain image animation with motion guidance, 2023.

[16] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

[17] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023.

[18] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024.

[19] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022.

[20] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[21] 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. *arXiv preprint arXiv:2402.00769*, 2024.

[22] Shanchuan Lin and Xiao Yang. Animatediff-lightning: Cross-model diffusion distillation. *arXiv preprint arXiv:2403.12706*, 2024.

[23] 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.

[24] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252. PMLR, 2023.

[25] Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8196–8206, 2024.

[26] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024.

[27] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[28] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*, 2024.

[29] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

[30] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net, 2019.

[31] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[32] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[33] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[34] 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.

[35] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023.

[36] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[37] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22930–22941, 2023.

[38] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2022.

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

[40] 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.

[41] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[42] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36, 2024.

[43] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

[44] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[45] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023.

[46] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *arXiv preprint arXiv:2404.13686*, 2024.

[47] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

[48] Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation. *arXiv preprint arXiv:2402.19159*, 2024.

[49] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[50] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

 https://doi.org/10.52202/079017-3290

[51] Jonas Kohler, Albert Pumarola, Edgar Schönfeld, Artsiom Sanakoyeu, Roshan Sumbaly, Peter Vajda, and Ali Thabet. Imagine flash: Accelerating emu diffusion models with backward distillation. *arXiv preprint arXiv:2405.05224*, 2024.

[52] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[53] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In *International conference on machine learning*, pages 30105–30118. PMLR, 2023.

[54] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021.

[55] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[56] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on image processing*, 6(2):298–311, 1997.

[57] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[58] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[59] Rohan Anil, Vineet Gupta, Tomer Koren, and Yoram Singer. Memory efficient adaptive optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[60] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[61] Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. On the content bias in fréchet video distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[62] 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.

[63] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main contributions and scope have been accurately summarized in the last paragraph of introduction. In Sec. 4, we conduct comprehensive experiments supporting the main claims made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Sec. 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-3290

Answer: [NA]

Justification: We do not introduce theoretical results in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the hyper-parameters we used in Sec. 4. We also include the description of the data we used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: As we mentioned in the abstract, we plan to release the code and pre-trained models with sufficient instructions to faithfully reproduce our results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the training and test details in Sec. 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Following existing works, we report FVD for our experiments, which does not include error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

 https://doi.org/10.52202/079017-3290

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computer resources information in Sec. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our work conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of our work in Sec. 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss this in the broader impacts.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Our experiments involve models from Stable Video Diffusion [13] and AnimateLCM [21]. Both works are properly cited in the paper. We also specially mention the version of each model we use in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

https://doi.org/10.52202/079017-3290

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code and pre-trained models are well documented and will be provided together.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.