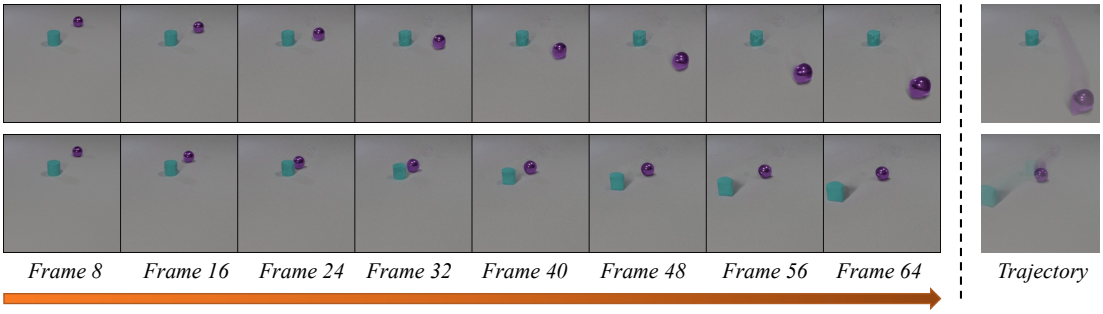
VIDM: Video Implicit Diffusion Models

Kangfu Mei, Vishal M. Patel

Johns Hopkins University
https://kfmei.page/vidm/



(a) Generated two videos with the same content on CLEVRER for collision reasoning

(b) Generated extra long SKY Time-lapse video with 128 frames (4 frames skipped)

(c) Generated high-resolution 256x256 TaiChi videos with 16 frames

Figure 1: Sample results corresponding to our method on multiple video datasets.

Abstract

Diffusion models have emerged as a powerful generative method for synthesizing high-quality and diverse set of images. In this paper, we propose a video generation method based on diffusion models, where the effects of motion are modeled in an implicit condition manner, i.e. one can sample plausible video motions according to the latent feature of frames. We improve the quality of the generated videos by proposing multiple strategies such as sampling space truncation, robustness penalty, and positional group normalization. Various experiments are conducted on datasets consisting of videos with different resolutions and different numbers of frames. The results show that the proposed method outperforms the state-of-the-art generative adversarial network-based methods by a significant margin in terms of FVD scores as well as perceptible visual quality.

Introduction

Image generation has gained significant traction since the introduction of Generative Adversarial Networks (GANs) (Goodfellow et al. 2014). In these methods, the idea is to generate new images that conform to the training data distribution. Following the success of image synthesis, video generation has also gained significant attention. Various video generation methods have been proposed in the literature including GAN-based methods (Vondrick, Pirsiavash, and Torralba 2016; Saito, Matsumoto, and Saito 2017; Tulyakov et al. 2018; Yu et al. 2022), Autoregressive models (Weissenborn, Täckström, and Uszkoreit 2020), and Time-series models (Tian et al. 2021; Skorokhodov, Tulyakov, and Elhoseiny 2022). An advantage of some of these generative models is that they can learn to synthesize high-quality videos without requiring any labels. These generative models have been shown to be beneficial in various high-level recognition tasks (Srivastava, Mansimov, and Salakhudinov 2015; Vondrick, Pirsiavash, and Torralba 2016).

A GAN-based video generation model was proposed by Vondrick, Pirsiavash, and Torralba (2016), which makes use of a spatio-temporal convolutional architecture and untangles the scene's foreground from the background. Another work proposed by Tulyakov et al. (2018) is a continuous-time video generator. In this method, a video is decomposed

into the content and motion vectors at generation and discriminated coherently by the discriminators. While these GAN-based methods can model plausible moving objects and scenes, a better video generation model should be able to model the distribution of internal spatial and temporal changes with regard to the video content.

Different from GANs, diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) as well as score-based models (Song et al. 2020) that model the probability directly have emerged as the new state-of-the-art generative models, and they have been shown to outperform GANs in various generation tasks (Dhariwal and Nichol 2021). By learning to reverse the diffusion process that adds noise into data in finite successive steps, diffusion models can gradually map a Gaussian distribution to the probability distribution corresponding to a real complex high-dimensional dataset. In its denoising process, conditional features like class labels of data can be applied to the network for specializing its sampling process. By appropriately using conditional features, diffusion models have shown impressive performance in various applications, *e.g.*, image deblurring (Whang et al. 2022) that conditions on the image residual, high-resolution image generation (Ho et al. 2022) that conditions on the low-resolution images, and image editing (Choi et al. 2021) that conditions on the style. With more expressive conditional features like CLIP embeddings (Radford et al. 2021), diffusion models like DALLE-2 (Ramesh et al. 2022) are capable of generating highly creative images with impressive photorealism. But the condition mechanism in diffusion models is non-trivial and requires careful design to improve the quality of the generated images.

We assume that the subspace of a real video can be represented as a subspace of the video content, and the video motion is then generated by traversing point on the video content subspace. Accurately modeling the content subspace increases the realism of frames, while accurately modeling the subspace of the trajectory regarding the video content can produce continuous and smooth video. Thus a better video generation model should own delicate modulation capability for simulating both the trajectory and realistic content.

Following this idea, we propose to model the video content and motion with two diffusion models separately. The first video frame is generated by the content generator. Subsequently, the motion generator generates the next video frame based on the latent map of the first frame and latest frame, i.e., an optical-flow like feature between the first and the latest frame estimated by an additional network. This enables implicitly modeling of dynamics by conditioning on the latent features. After training, the optimized condition can best represent the spatial and temporal changes for generating the next frame. By iteratively running the motion generator, the final video is generated in an autoregressive manner. We experimentally find that the estimated condition significantly enhances the modeling capability of diffusion models, Such an expressive model is capable of simulating the trajectory of videos according to the conditional latent.

The major idea of our video implicit diffusion models is:

- *Content Generator:* We propose to learn video content separately with an introduced diffusion model on video frames. It simplifies video generation modeling and provides easy scalability of complex models. Two heuristic mechanisms, including constant truncation and robustness penalty, are proposed for further improving its performance.
- *Motion Generator:* We propose a motion generator for modeling spatial and temporal changes. It can generate future frames according to the generated content in an autoregressive way. The generator is implicitly conditioned on the latent code predicted by a module similar to an optical-flow network. Furthermore, the coherency of spatial and temporal changes is regularized with an introduced positional group normalization, and the learning is simplified with our proposed adaptive feature residual.

The effectiveness of the proposed model is demonstrated on various datasets by comparing the performance with several state-of-the-art works, including very recent works MoCoGAN-HD (Tian et al. 2021), DIGAN (Yu et al. 2022), and StyleGAN-V (Skorokhodov, Tulyakov, and Elhoseiny 2022). It is shown that our method achieves significantly better quantitative performance of Fréchet video distance and is experimentally observed to be capable of generating more realistic results.

Denoising Diffusion Probabilistic Model

Based on the success of diffusion-based models, we extend the generation process from 2D images into 3D videos and keep the modification as minimal as possible. Our approach is based on Denoising Diffusion Probabilistic Mdoel (DDPM) proposed by Ho, Jain, and Abbeel (2020) and its variant Guided-DDPM from Dhariwal and Nichol (2021).

Learning Process. DDPM models the distribution of images $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ in a denoising process, and it learns noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with respect to timesteps t (out of T) and defines noisy image $\mathbf{x}_t$ as a function $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, which is implemented as a modified U-Net (Salimans et al. 2017) $\boldsymbol{\epsilon}_\theta(\cdot)$ with parameters θ. A simplified learning objective is

$$\mathcal{L}_\theta(\epsilon, \mathbf{x}_t, t) = \|\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \boldsymbol{\epsilon}\|. \tag{1}$$

Noise Definition. Various attempts have been made to improve the form of $\mathbf{x}_t$. The basic formulation comes by combining the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with a clean image $\mathbf{x}_0$ in t steps according to some pre-defined noise schedules α_t and its variant $\bar{\alpha}_t$ as $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$.

Inference Process. The generation process starts from noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with random noise $\boldsymbol{\epsilon}$ and predefined variance σ_t and is defined as

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t\boldsymbol{\epsilon}. \tag{2}$$

Truncation Trick. Compared with the GAN-based methods, though the superiority in the diversity of generation is achieved, diffusion models often result in the generation of poor quality of objects. Inspired by the heavily explored approach used by GANs, that is sampling from a truncated or a shrunk sampling space (Brock, Donahue, and Simonyan 2018; Kingma and Dhariwal 2018; Karras, Laine, and Aila 2019), we propose to truncate noise $\boldsymbol{\epsilon}$ implicitly.

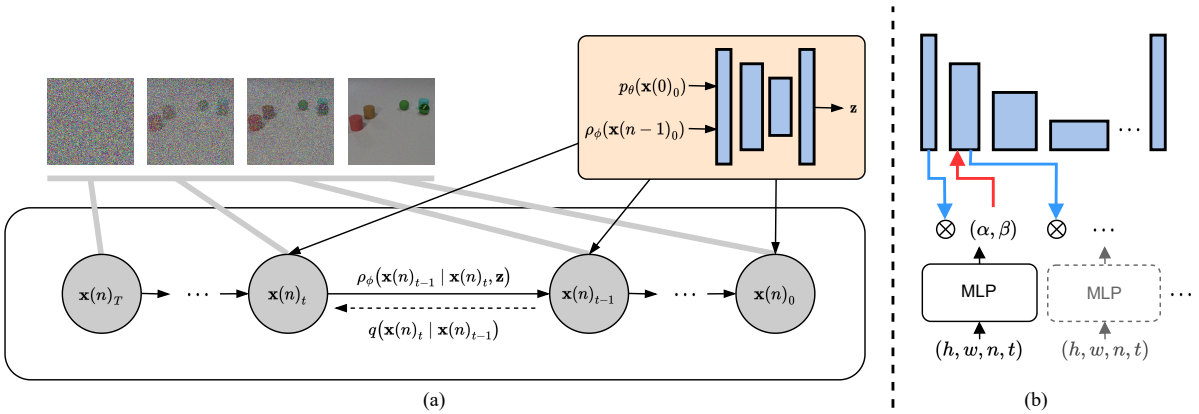


Figure 2: (a) Illustration of our graphical model at the n-th video frame sampling process. (b) The proposed positional group normalization concept when it is applied to the diffusion network.

Inspired by the practice of StyleGAN (Karras, Laine, and Aila 2019), which starts from a learnable constant and then gradually upsamples the features until the final output layer, we propose to concatenate noisy image $\mathbf{x}_t$ with a learnable constant c that has the same dimension as $\mathbf{x}_t$ at each diffusion step. Such a strategy truncates the sampling space of the noise in an implicit way without modifying the network architecture, and the learning objective is slightly changed as $\|\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, c, t) - \boldsymbol{\epsilon}\|$. During inference, the constant c is fixed and the inference process equation 2 is subsequently updated as

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, c, t)\right) + \sigma_t\boldsymbol{\epsilon}. \quad (3)$$

Robustness Penalty. Dropout layer has been empirically applied in DDPM for suppressing overfitting artifacts. However, the practice of applying dropout depends on the dataset and harms the general performance in most cases. We observed that overfitting not only depends on the dataset but different classes of the same dataset as well and thus dropout is conventionally avoided.

To enable an adaptive strategy for preventing overfitting, we propose to add a penalty function (Charbonnier et al. 1994) at the learning objective instead of dropout layers as

$$\mathcal{L}_\theta(\boldsymbol{\epsilon}, \mathbf{x}_t, t) = \sqrt{(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, c, t) - \boldsymbol{\epsilon})^2 + \eta^2}, \quad (4)$$

where η is a constant that is experimentally set as $1e-8$, while the other settings, including $\{1e-5, 1e-6, 1e-7\}$, haven't shown significantly better performance. Such a modification doesn't hurt the differentiability of the original learning objective.

Video Implicit Diffusion Model

Our proposed video generation method consists of two streams for content and motion generation, respectively. The two streams share a similar network architecture but different in learning objectives. In addition, they have different conditions which helps to keep the design redundancy minimal and reduces the optimization cost. We denote the n-th frame of the N-frame video as $\mathbf{x}(n)_0$. The noisy frame at the tth timestep is denoted as $\mathbf{x}(n)_t$.

Content Generator models the distribution of random video frames $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ with a network $\boldsymbol{\epsilon}_\theta(\cdot)$ and is truncated by constant tensor c. The frame $\mathbf{x}_0$ is randomly selected from videos without specification. The network $\boldsymbol{\epsilon}_\theta(\cdot)$ is the modified U-Net proposed by Dhariwal and Nichol (2021) with *Multi-Head Attention* (Vaswani et al. 2017) and utilizes *GroupNorm* (Wu and He 2018).

Motion Generator models the distribution of motion from the first frame to the random n-th frame, and it is implemented with another network $\rho_\phi(\cdot)$ with parameters ϕ. Therefore, the learning process minimizes the difference between $\boldsymbol{\rho}_\phi(\mathbf{x}(0)_0, n)$ and $\mathbf{x}(n)_0$ as Figure 2 shows, which is similar to recent implicit neural function methods (Yu et al. 2022; Skorokhodov, Tulyakov, and Elhoseiny 2022).

By experimentally combining the two streams, one can model video data. However, due to the complexity of video data, we observed that the basic implementation does not converge. In addition, it losses significant generation quality and generates discontinuous motions. Therefore, we propose to extend the aforementioned video generation process with the following improvements.

Positional Group Normalization

Our first key idea for improving the diffusion network is to incorporate the spatial and temporal positional encoding of 4D coordinates (h, w, n, t) between each U-Net blocks, for modeling continuous changes in both the space h, w and time n with different diffusion timesteps t. The correlation between spatial and temporal features crucially affects the continuity of video data but is conventionally ignored due to its complexity. Empirically, such complexity can be decomposed for modeling in an iterative denoising process. We propose to directly incorporate the correlation into networks in a feature modulation manner, similar to AdaIN (Karras, Laine, and Aila 2019) and FiLM (Perez et al. 2018).

The concept is illustrated in the right part of Figure 2. Specifically, the positional encoding mapped from 4D coordinates is extracted through an MLP (fully-connected neural network) with sinusoidal activation (Sitzmann et al. 2020) after its first layer. Recent studies on implicit neural representations (INRs) (Sitzmann et al. 2020; Tancik et al. 2020)

have shown that periodic activation is capable of modeling high dimensional space with coordinates. Inspired by it, our introduced Positional Group Normalization (PosGN) based on group-norm (Wu and He 2018) is defined as

$$\alpha, \beta = \text{MLP}(h, w, n, t) \tag{5}$$

$$\text{PosGN}(x, \alpha, \beta) = \alpha \cdot \text{GroupNorm}(x) + \beta, \tag{6}$$

where x is the obtained feature from the U-Net blocks, (α, β) is a pair of affine transformation parameters extracted from the MLP, and it then scales and shifts feature x using parameters (α, β). PosGN is based on the empirical superiority of adaptive group normalization (AdaGN) (Nichol and Dhariwal 2021), which has been shown to benefit diffusion models, and the difference between them are the introduced periodic activated MLP and the additional spatial and temporal dimensions. Compared with the recent INR-based work, PosGN is particularly suited for diffusion models. It is because the noisy images are essential conditions that cannot be replaced by coordinates as INRs have done. Besides, PosGN provides a hierarchical feature modulation when it is incorporated into the applied diffusion networks.

As a result, our proposed VIDM benefits from the capability of modeling spatial and temporal changes led by PosGN. Based on the new paradigm, the learning objective of our motion generation extended from the content modeling equation 4 for an arbitrary n-th frame is formulated as

$$\mathcal{L}_\phi(\boldsymbol{\epsilon}, \mathbf{x}(n)_t, t, n) = \sqrt{(\boldsymbol{\rho}_\phi(\mathbf{x}(n)_t, t, n) - \boldsymbol{\epsilon})^2 + \eta^2}. \tag{7}$$

Coordinates (h, w) are derived from features on-the-fly and thus are not treated as the network input. For convenience and efficiency, coordinates (h, w, n, t) are only generated at the first time and then cached for the next running. Therefore, PosGN shares a very similar computational cost as the vanilla AdaGN when the running times are large, which is natural to diffusion models. In the rest of this paper, we use PosGN as our default settings and denote $\boldsymbol{\rho}_\phi(\cdot, t, n)$ as $\boldsymbol{\rho}_\phi(\cdot)$ for simplification.

Implicit Motion Condition

Modeling long continuous video data has been a long-standing problem, even though we have seen the exploration in INRs and our proposed PosGN with positional encoding, the intermediate information between long video frames cannot be accurately represented. Furthermore, from the results in the literature (Yu et al. 2022; Skorokhodov, Tulyakov, and Elhoseiny 2022), we find that the intermediate information plays a crucial role in the video continuation, otherwise, the generated long videos only contain nearly meaningless motions.

Our second idea is extended from the proposed PosGN, based on the time condition, instead of explicit coordinates, we propose to condition on the latent code of the latest frame and the first frame at the denoising process. The latent code is an optical flow (Horn and Schunck 1981) like feature estimated by an additional network $\mathbf{v}(\cdot)$, implemented as SpyNet (Ranjan and Black 2017), which has been demonstrated in motion extraction for video enhancement and interpolation. To elaborate, a pretrained optical flow estimation network $\mathbf{v}(\cdot)$ is applied to estimate the latent $\mathbf{z}$ between

Algorithm 1: Motion Learning

input: random frames $\{\mathbf{x}(0), \mathbf{x}(n-1), \mathbf{x}(n)\}$
repeat
  $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{z} = \mathbf{v}(\mathbf{x}(0)_0, \mathbf{x}(n-1)_0)$
  $r = \hat{\boldsymbol{\rho}}_\phi(\mathbf{x}(0)_0)$
  $\mathbf{x}(n)_t = \sqrt{\bar{\alpha}_t}\mathbf{x}(n)_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
  Take gradient descent step on
    $\nabla_\phi \sqrt{(\boldsymbol{\epsilon} - \boldsymbol{\rho}_\phi(\mathbf{x}(n)_t, \mathbf{z}) - r)^2 + \eta^2}$
until converged
return: motion network $\boldsymbol{\rho}_\phi(\cdot)$

Algorithm 2: Video Generation

for $n = 0, \dots, N-1$ **do**
  $\mathbf{x}(n)_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $t = T, \dots, 1$ **do**
    if $n = 0$
      $\mathbf{x}(n)_{t-1} \sim \boldsymbol{\epsilon}_\theta(\mathbf{x}(n)_t, c, t)$
    else
      $\mathbf{z} = \mathbf{v}(\mathbf{x}(0)_0, \mathbf{x}(n-1)_0)$
      $r = \hat{\boldsymbol{\rho}}_\phi(\mathbf{x}(0)_0)$
      $\mathbf{x}(n)_{t-1} \sim \boldsymbol{\rho}_\phi(\mathbf{x}(n)_t, \mathbf{z}) + r$
  end for
end for

frames $\mathbf{v}(\mathbf{x}(0)_0, \mathbf{x}(n-1)_0)$ for the n-th frame $\mathbf{x}(n)_0$ generation, which is performed in an autoregressive manner. Since the latent code is capable of ensembling the continuous motion feature that consistently exist in the denoising process, it can ensure that the intermediate information is implicitly incorporated into learning. Therefore, the learning process with the implicit latent condition is

$$\mathbf{z} = \mathbf{v}(\mathbf{x}(0)_0, \mathbf{x}(n-1)_0) \tag{8}$$

$$\mathcal{L}_\phi(\boldsymbol{\epsilon}, \mathbf{x}(n)_t, \mathbf{z}) = \sqrt{(\boldsymbol{\rho}_\phi(\mathbf{x}(n)_t, \mathbf{z}) - \boldsymbol{\epsilon})^2 + \eta^2}. \tag{9}$$

The parameters of $\mathbf{v}(\cdot)$ are updated with the diffusion networks together without specification. The cost of conditioning on the latent at each denoising process is only increased at the first timesteps and can be then cached. As will be shown in the ablation study, implicit learning is crucial for modeling long video data and can significantly improve the ultimate performance.

Adaptive Feature Residual

To further simplify the motion modeling complexity, we propose to model the residual of content features at each denoising timestep adaptively. An additional encoder that shares the similar architecture of the diffusion network is utilized, and it conditions on the first frame $\mathbf{x}(0)_0$ and timesteps t. We denote the encoding as $\hat{\rho}_\phi(\cdot)$ and the residual feature as r, and thus network $\rho_\phi(\cdot)$ is actually learning to synthesize the residual, which significantly simplifies the learning at each timestep and enables better implicit motion learning.

The complete procedure of our method for both motion learning and video generation is detailed in Algorithm 1.

Remark that content generation learning is kept the same as DDPM except for the truncation trick and robustness penalty is applied for enhancing the generation capability.

Experiments

Datasets and settings. Most datasets follow the protocols of their original papers except where specified. To compare the visual quality of the results, we use the I3D network trained on Kinetics-400 (Kay et al. 2017) for reporting the Fréchet video distance (FVD) (Unterthiner et al. 2018) performance, which measures the probability distribution difference between two groups of video results and is recognized by the other prior arts (Yu et al. 2022; Skorokhodov, Tulyakov, and Elhoseiny 2022). For reference, we also report the Inception score (IS) (Salimans et al. 2016) performance and Fréchet inception distance (FID) (Heusel et al. 2017) following the evaluation procedure of DIGAN (Yu et al. 2022). All evaluation is conducted on 2048 randomly selected real and generated videos for reducing variance. The experiments are conducted on *UCF-101* (Soomro, Zamir, and Shah 2012), *TaiChi-HD* (Siarohin et al. 2019), *Sky Time-lapse* (Xiong et al. 2018), and *CLEVRER* (Yi et al. 2020).

Baselines. The major baseline for comparison is DIGAN (Yu et al. 2022), which is the current state-of-the-art in video generation and is the first work that incorporates INRs. We also compare the performance of our method with that of VGAN (Vondrick, Pirsiavash, and Torralba 2016), TAGN (Saito, Matsumoto, and Saito 2017), MoCoGAN (Tulyakov et al. 2018), ProgressiveVGAN (Acharya et al. 2018), DVD-GAN (Clark, Donahue, and Simonyan 2019), LDVD-GAN (Kahembwe and Ramamoorthy 2020), TGANv2 (Saito et al. 2020), MoCoGAN-HD (Tian et al. 2021), VideoGPT (Yan et al. 2021), StyleGAN-V (Skorokhodov, Tulyakov, and Elhoseiny 2022), VDM (Ho et al. 2022), and TATS (Ge et al. 2022). We collect the performance score from the references or re-implemented results from DIGAN and StyleGAN-V if available. For the CLEVRER performance, we train DIGAN and StyleGAN-V with their official code and our implementation with the same settings.

Diffusion Network. The diffusion network architecture of our method is an autoencoder network that follows the design of PixelCNN++ (Salimans et al. 2017). We apply multiple multi-head attention modules (Vaswani et al. 2017) at features in a resolution of 16×16 for capturing long-range dependence that benefits the perceptual quality. It has been verified by DDPM (Ho, Jain, and Abbeel 2020) and its variants (Dhariwal and Nichol 2021; Nichol and Dhariwal 2021), and we keep minimal changes.

Main Results. We present the main quantitative results comparison in Table 1 and Table 2, and the main qualitative results comparison is Figure 3. We remark that our performance significantly outperforms the very recent state-of-the-art DIGAN and StyleGAN-V in all of the video data as can be seen from the two tables. Among them, *128-TaiChi* and *256-UCF101* is the hardest video data since their movement is minimal and Frames Per Second (FPS) is varying between

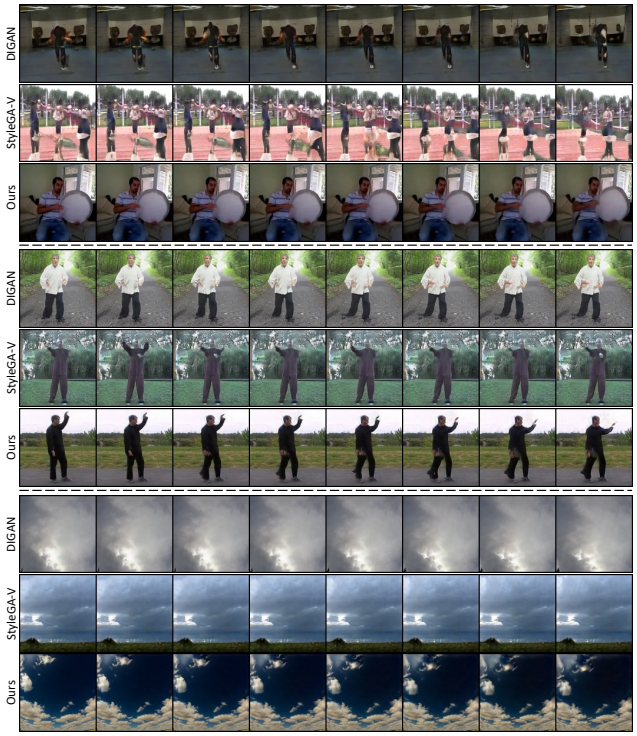


Figure 3: Sample result comparisons on the *256-UCF101*^{16}, *128-TaiChi*^{16}, and *256-SkyTimelapse*^{16} datasets. Each presented frame is selected with 2 frames interval.

videos, but our method can still achieve comparable performance and even better without discriminators.

Ablations. Multiple potential design choices are available in our final method, and most of them affect the results to some degree. We ablate the core components and show the details in Table 3a for content generator ablations and Table 3b for motion generator ablations. As the results are shown in Table 3a, the removed sampling space truncation and robustness penalty hurt the performance of content modeling. These results also verify that removing the robustness penalty decreases both the content modeling ability and motion modeling ability.

For the motion generator, we measure the ablation effects by comparing generated videos in a varying number of frames, which is the most representative score for measuring continuous and smoothness differences. In Table 3b, we remove the positional group normalization and implicit motion conditions to see the difference. It is surprising that the modeling capability severely depends on the two proposed components, especially for long video generation. From the results visualized in Figure 3c, we can notice that simply applying diffusion models (i.e., *Ablation1*) without modification can only generate static images. Applying implicit conditions without PosGN (i.e., *Ablation2*) faces the same issue since they cannot model the spatial and temporal changes. In contrast, even though applying PosGN without implicit conditions (i.e., *Ablation3*) can help the network generates different frames, its results are still noncontinuous. In Figure 4, we visualize the latent and its corresponding video

	MoCoGAN† *CVPR18*	MoCoGAN-HD *ICLR21*	VideoGPT *arXiV21*	DIGAN *ICLR22*	DIGAN‡ *ICLR22*	StyleGAN-V *CVPR22*	TATS *ECCV22*	VIDM (ours)
256-UCF101^{16}	1821.4	1729.6	2880.6	1630.2	471.9	1431.0	332	**294.7**
256-UCF101^{128}	2311.3	2606.5	*N/A*	2293.7	*N/A*	1773.4	-	**1531.9**
256-SkyTimelapse^{16}	85.9	164.1	222.7	83.1	83.1	79.5	132	**57.4**
256-SkyTimelapse^{128}	272.8	878.1	*N/A*	196.7	196.7	197.0	-	**140.9**

	DIGAN	StyleGAN-V	VIDM (ours)
256-CLEVRER^{16}	112.5	106.1	**87.4**
256-CLEVRER^{128}	531.7	493.3	**426.5**

	DIGAN	StyleGAN-V	VIDM (ours)
128-TaiChi^{16}	128.1	143.5	**121.9**
128-TaiChi^{128}	748.0	691.1	**563.6**

Table 1: Fréchet video distance (Unterthiner et al. 2018) comparison. The compared methods are re-trained on the CLEVRER dataset by us, and by Skorokhodov, Tulyakov, and Elhoseiny (2022) and Yu et al. (2022) on the other datasets with their official implementation. MoCoGAN† is implemented with StyleGAN2 as its backbone. DIGAN‡ is class conditional.

	Train split							
	VGAN *NeurIPS16*	TGAN *ICCV17*	MoCoGAN *CVPR18*	ProgressiveVGAN *arXiv18*	LDVD-GAN *NN20*	VideoGPT *arXiv21*	TGANv2 IJCV20	DIGAN *ICLR22*
128-UCF101^{16} *IS (↑)*	8.31$\pm$.09	11.85$\pm$.07	12.42$\pm$.07	14.56$\pm$.05	22.91$\pm$.19	24.69$\pm$.30	28.87$\pm$.67	29.71$\pm$.53
128-UCF101^{16} *FID (↓)*	-	-	-	-	-	-	1209$\pm$28	655$\pm$22

	Train split		*Train+test split*						
	VIDM *Ours*	VIDM† *Ours*	DVD-GAN *arXiV19*	MoCoGAN-HD *ICLR21*	DIGAN *ICLR22*	StyleGAN-V *CVPR22*	DIGAN‡ *ICLR22*	VDM *arXiv22*	VIDM† Ours
128-UCF101^{16} *IS (↑)*	*53.34*	35.20	27.38$\pm$.53	32.36	32.70$\pm$.35	32.70$\pm$.35	*59.68$\pm$.45*	57$\pm$.62	**64.17**
128-UCF101^{16} *FID (↓)*	306	*471*	-	838	577$\pm$21	-	-	*295$\pm$3*	**263**

Table 2: IS and FVD comparisons. For fair comparisons, we re-train our VIDM without video class condition, named VIDM†.

frames for further clarification.

Related Work

Generative Models. Existing generative models can be categorized into likelihood-based and implicit models, based on the way of representing probability distribution. Among them, Variational Auto-encoders (VAEs) (Kingma and Welling 2013), Autoregressive models (Van Oord, Kalchbrenner, and Kavukcuoglu 2016; Germain et al. 2015), Normalizing Flow (Dinh, Sohl-Dickstein, and Bengio 2016), and Diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) directly model the probability distribution of data via maximum likelihood. In contrast, GANs (Goodfellow et al. 2014) implicitly represent the probability distribution via their sampled results. Though the idea of GANs is simple, the boundary has been significantly pushed by GANs and their representative variants, including StyleGAN (Karras, Laine, and Aila 2019; Karras et al. 2020,) and BigGAN (Brock, Donahue, and Simonyan 2018). Moreover, many general techniques based on GANs have emerged, including R_1 regularization (Mescheder, Geiger, and Nowozin 2018), path length regularization (Karras et al. 2020), truncation trick (Karras, Laine, and Aila 2019), spectral normalization (Miyato et al. 2018), image inversion (Mei and Patel 2021), and adaptive discriminator (Karras et al. 2020). However, we find that such techniques are rarely explored in diffusion models. In this paper, inspired by these heavily refined techniques, we introduce several adaptive methods for refining diffusion models and observe that they benefit for both video and image generation.

Conditional Generative Models. Modeling the probability distribution of complex datasets such as ImageNet (Deng et al. 2009) can face potential training instability and mode collapse issues. Therefore, the way of leveraging additional conditions as a guidance is explored and becoming the most promising way of mitigating the issues. For GANs, class information can be fed into the generator (Mirza and Osindero 2014; Odena, Olah, and Shlens 2017; De Vries et al. 2017; Dumoulin, Shlens, and Kudlur 2016; Brock, Donahue, and Simonyan 2018) and the discriminator (Miyato and Koyama 2018; Karras, Laine, and Aila 2019) for fascinating class-conditional sampling. For diffusion models, the class condition shows a better performance boost as the class embeddings used in DDPM (Ho, Jain, and Abbeel 2020). Furthermore, resulting from the iterative denoising process of diffusion models, which enables hierarchical conditional features, utilizing the class feature of noisy images of different time steps can help diffusion models achieve the new art (Dhariwal and Nichol 2021). Different from class conditions, the modality of conditions could be images (Ledig et al. 2017; Nair, Mei, and Patel 2022) and even texts like DALLE (Ramesh et al. 2021) for different aims. VIDM is the first work that explores the implicit conditions for video generation, and it also benefits from the iterative denoising process of diffusion models, which allows hierarchical conditional features of complex spatial-temporal changing.

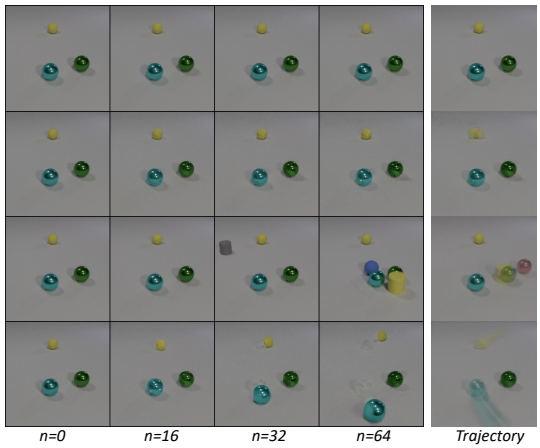
Video Generation. Video generation has been dominated by 3D CNNs (Tran et al. 2015) for a long time until the recent emergence of Implicit Neural Representation (INR) (Sitzmann et al. 2020; Tancik et al. 2020). The early 3D CNNs based video generation works take all frames of the video as a single point on the video subspace. They then generate a cuboid as the result of each sampling process (Vondrick, Pirsiavash, and Torralba 2016; Saito, Mat-

	FID	IS	FVD^{16}
vanilla one	23.0	3.04	115.4
w/o *sampling space truncation*	21.1	3.07	107.9
w/o *robustness penalty*	19.4	3.07	95.5
default VIDM	18.4	3.07	87.4

(a) Ablation study regarding content generator.

	FVD^{16}	FVD^{64}	FVD^{128}
vanilla one	603.7	610.0	648.7
w/o *PosGN*	532.1	581.3	604.5
w/o *Implicit Conditions*	584.8	552.1	614.1
default VIDM	87.4	286.6	426.5

(b) Ablation study regarding motion generator.



(c) Ablation results in different settings.

Table 3: Ablations on different settings with quantitative and qualitative results.

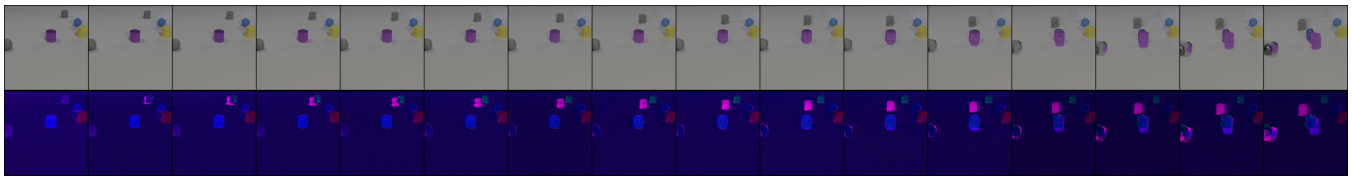


Figure 4: Visualization of the generated latent and its corresponding frames.

sumoto, and Saito 2017), and such a manner has been extended into the recent diffusion fashion (Ho et al. 2022; Harvey et al. 2022). However, this line can hardly achieve desired results due to the difficulty of modeling spatial-temporal changing, and their scalability is significantly limited according to the cubic complexity (Saito et al. 2020). Later work decomposes the generation process into content and motion separately (Tulyakov et al. 2018; Clark, Donahue, and Simonyan 2019; Tian et al. 2021; Fox et al. 2021), which simplifies the learning but still requires the discriminator to apply 3D CNNs on extracting temporal features. The other line of video generation (Yu et al. 2022; Skorokhodov, Tulyakov, and Elhoseiny 2022) based on INRs is similar to the image generation applications work (Skorokhodov, Ignatyev, and Elhoseiny 2021), which adds additional temporal dimension at the coordinates and thus can process each frame separately. However, such an INR protocol can hardly be applied to diffusion models. Therefore, our method incorporates the coordinate embeddings of INRs as the normalization and conditions on the implicit latent. We experimentally find that the new paradigm benefits the continuous of the generated complex videos.

Limitations and Ethics Statement. The major limitation of this work comes from the efficiency issue of diffusion models. Limited by the expressibility of the Gaussian process, multiple iterative denoising process is required before producing plausible results. Therefore, the complexity of video generation consists of the number of video frames and the number of diffusion time steps. The potential negative societal impacts of this work come from the generated unethical videos. These generated videos a.k.a Deepfake have emerged as an important social issue and attracted great attention. However, we are happy to see that significant funds and efforts have been devoted to detecting these fake videos, including DARPA's Semantic Forensics program which is highly inspired by the StyleGAN series (Karras, Laine, and Aila 2019). Our work can be useful in promoting them.

Conclusion

In this work, we proposed a new diffusion probabilistic model for video data, which provides a unique implicit condition paradigm for modeling continuous spatial-temporal changing of videos. The model is capable of sampling frames according to latent that encodes dynamics. Comprehensive experiments on the high-resolution, long video data demonstrated our method not only with visual quality superiority but also better diversity. We hope the work would benefit and inspire both video generation and conditional diffusion models as a strong baseline in the future.

Acknowledgement

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-21-2-0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. We thank Meihan Wei for helpful feedbacks.

References

[1] Acharya, D.; Huang, Z.; Paudel, D. P.; and Van Gool, L. 2018. Towards high resolution video generation with progressive growing of sliced wasserstein gans. *arXiv preprint arXiv:1810.02419*.

[2] Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

[3] Charbonnier, P.; Blanc-Feraud, L.; Aubert, G.; and Barlaud, M. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*.

[4] Choi, J.; Kim, S.; Jeong, Y.; Gwon, Y.; and Yoon, S. 2021. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *ICCV*.

[5] Clark, A.; Donahue, J.; and Simonyan, K. 2019. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*.

[6] De Vries, H.; Strub, F.; Mary, J.; Larochelle, H.; Pietquin, O.; and Courville, A. C. 2017. Modulating early visual processing by language. In *NeurIPS*.

[7] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

[8] Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. In *NeruIPS*.

[9] Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

[10] Dumoulin, V.; Shlens, J.; and Kudlur, M. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*.

[11] Fox, G.; Tewari, A.; Elgharib, M.; and Theobalt, C. 2021. Stylevideogan: A temporal generative model using a pretrained stylegan. *arXiv preprint arXiv:2107.07224*.

[12] Ge, S.; Hayes, T.; Yang, H.; Yin, X.; Pang, G.; Jacobs, D.; Huang, J.-B.; and Parikh, D. 2022. Long video generation with time-agnostic vqgan and time-sensitive transformer. *arXiv preprint arXiv:2204.03638*.

[13] Germain, M.; Gregor, K.; Murray, I.; and Larochelle, H. 2015. Made: Masked autoencoder for distribution estimation. In *ICML*.

[14] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*.

[15] Harvey, W.; Naderiparizi, S.; Masrani, V.; Weilbach, C.; and Wood, F. 2022. Flexible Diffusion Modeling of Long Videos. *arXiv preprint arXiv:2205.11495*.

[16] Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*.

[17] Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

[18] Ho, J.; Saharia, C.; Chan, W.; Fleet, D. J.; Norouzi, M.; and Salimans, T. 2022. Cascaded diffusion models for high fidelity image generation. *JMLR*.

[19] Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022. Video diffusion models. *arXiv preprint arXiv:2204.03458*.

[20] Horn, B. K.; and Schunck, B. G. 1981. Determining optical flow. *Artificial intelligence*, 17(1-3): 185–203.

[21] Kahembwe, E.; and Ramamoorthy, S. 2020. Lower dimensional kernels for video discriminators. *Neural Networks*.

[22] Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; and Aila, T. 2020. Training generative adversarial networks with limited data. In *NeurIPS*.

[23] Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *CVPR*.

[24] Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *CVPR*.

[25] Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

[26] Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*.

[27] Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[28] Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*.

[29] Mei, K.; and Patel, V. M. 2021. Ltt-gan: Looking through turbulence by inverting gans. *arXiv preprint arXiv:2112.02379*.

[30] Mescheder, L.; Geiger, A.; and Nowozin, S. 2018. Which training methods for GANs do actually converge? In *ICML*.

[31] Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

[32] Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

[33] Miyato, T.; and Koyama, M. 2018. cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637*.

[34] Nair, N. G.; Mei, K.; and Patel, V. M. 2022. AT-DDPM: Restoring Faces degraded by Atmospheric Turbulence using Denoising Diffusion Probabilistic Models. In *WACV*.

[35] Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *ICML*.

[36] Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *ICML*.

[37] Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *AAAI*.

[38] Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

[39] Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

[40] Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*.

[41] Ranjan, A.; and Black, M. J. 2017. Optical flow estimation using a spatial pyramid network. In *CVPR*.

[42] Saito, M.; Matsumoto, E.; and Saito, S. 2017. Temporal generative adversarial nets with singular value clipping. In *ICCV*.

[43] Saito, M.; Saito, S.; Koyama, M.; and Kobayashi, S. 2020. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *IJCV*.

[44] Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *NeurIPS*.

[45] Salimans, T.; Karpathy, A.; Chen, X.; and Kingma, D. P. 2017. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*.

[46] Siarohin, A.; Lathuilière, S.; Tulyakov, S.; Ricci, E.; and Sebe, N. 2019. First order motion model for image animation. In *NeurIPS*.

[47] Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; and Wetzstein, G. 2020. Implicit neural representations with periodic activation functions. In *NeurIPS*.

[48] Skorokhodov, I.; Ignatyev, S.; and Elhoseiny, M. 2021. Adversarial generation of continuous images. In *CVPR*.

[49] Skorokhodov, I.; Tulyakov, S.; and Elhoseiny, M. 2022. StyleGAN-V: A Continuous Video Generator with the Price, Image Quality and Perks of StyleGAN2. In *CVPR*.

[50] Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*.

[51] Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

[52] Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

[53] Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *ICML*.

[54] Tancik, M.; Srinivasan, P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.; and Ng, R. 2020. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*.

[55] Tian, Y.; Ren, J.; Chai, M.; Olszewski, K.; Peng, X.; Metaxas, D. N.; and Tulyakov, S. 2021. A good image generator is what you need for high-resolution video synthesis. *arXiv preprint arXiv:2104.15069*.

[56] Tian, Y.; Ren, J.; Chai, M.; Olszewski, K.; Peng, X.; Metaxas, D. N.; and Tulyakov, S. 2021. A Good Image Generator Is What You Need for High-Resolution Video Synthesis. In *ICLR*.

[57] Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

[58] Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018. Mocogan: Decomposing motion and content for video generation. In *CVPR*.

[59] Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2018. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*.

[60] Van Oord, A.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel recurrent neural networks. In *ICML*.

[61] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

[62] Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. In *NeurIPS*.

[63] Weissenborn, D.; Täckström, O.; and Uszkoreit, J. 2020. Scaling Autoregressive Video Models. In *ICLR*.

[64] Whang, J.; Delbracio, M.; Talebi, H.; Saharia, C.; Dimakis, A. G.; and Milanfar, P. 2022. Deblurring via Stochastic Refinement. In *CVPR*.

[65] Wu, Y.; and He, K. 2018. Group normalization. In *ECCV*.

[66] Xiong, W.; Luo, W.; Ma, L.; Liu, W.; and Luo, J. 2018. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *CVPR*.

[67] Yan, W.; Zhang, Y.; Abbeel, P.; and Srinivas, A. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*.

[68] Yi, K.; Gan, C.; Li, Y.; Kohli, P.; Wu, J.; Torralba, A.; and Tenenbaum, J. B. 2020. Clevrer: Collision events for video representation and reasoning. In *ICLR*.

[69] Yu, S.; Tack, J.; Mo, S.; Kim, H.; Kim, J.; Ha, J.-W.; and Shin, J. 2022. Generating Videos with Dynamics-aware Implicit Generative Adversarial Networks. In *ICLR*.