MV-Diffusion: Motion-aware Video Diffusion Model

Zijun Deng
Wangxuan Institute of Computer Technology & National Key Laboratory for Multimedia Information Processing,
Peking University
Beijing, China
dengzijun@stu.pku.edu.cn

Xiangteng He
Wangxuan Institute of Computer Technology & National Key Laboratory for Multimedia Information Processing,
Peking University
Beijing, China
hexiangteng@pku.edu.cn

Yuxin Peng[*]
Wangxuan Institute of Computer Technology & National Key Laboratory for Multimedia Information Processing,
Peking University
Beijing, China
pengyuxin@pku.edu.cn

Xiongwei Zhu
Kuaishou Technology
Beijing, China
zhuxiongwei@kuaishou.com

Lele Cheng
Kuaishou Technology
Beijing, China
chenglele@kuaishou.com

ABSTRACT

In this paper, we present a Motion-aware Video Diffusion Model (MV-Diffusion) for enhancing the temporal consistency of generated videos using autoregressive diffusion models. Despite the success of diffusion models in various vision generation tasks, generating high-quality and realistic videos with coherent temporal structure remains a challenging problem. Current methods have primarily focused on capturing implicit motion features within a restricted window of RGB frames, rather than explicitly modeling the motion. To address this, we focus on improving the temporal modeling ability of the current autoregressive video diffusion approach by leveraging rich temporal trajectory information in a global context and explicitly modeling local motion trends. The main contributions of this research include: (1) a **Trajectory Modeling (TM)** block that enhances the model's conditioning by incorporating global motion trajectory information, (2) a **Motion Trend Attention (MTA)** block that utilizes a cross-attention mechanism to explicitly infer motion trends from the optical flow rather than implicitly learning from RGB input. Experimental results on three video generation tasks using four datasets show the effectiveness of our proposed MV-Diffusion, outperforming existing state-of-the-art approaches. The code is available at https://github.com/PKU-ICST-MIPL/MV-Diffusion_ACMMM2023.

CCS CONCEPTS

• **Computing methodologies** → *Computer vision*; *Motion capture*.

KEYWORDS

Autoregressive video diffusion; Video generation; Motion capture

[*] Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612405

ACM Reference Format:
Zijun Deng, Xiangteng He, Yuxin Peng, Xiongwei Zhu, and Lele Cheng. 2023. MV-Diffusion: Motion-aware Video Diffusion Model. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3612405

1 INTRODUCTION

The field of video generation has witnessed significant advancements with the emergence of Generative Adversarial Networks (GANs) [10]. Recently, Denoising Diffusion Probabilistic Models (DDPM) [12] have emerged as powerful generative models and surpassed GANs in various synthesis tasks. With improved training stability and model generalizability, diffusion models have played a crucial role in the advancement of video generation research and have received growing attention.

However, generating high-fidelity videos remains a challenging task. Achieving realism and consistency in both spatial and temporal aspects of the video is crucial for a successful video generation model. In terms of spatial realism, the video should accurately depict the visual characteristics of the scene. Moreover, achieving temporal coherence in the object motion is also essential for a temporally consistent video. This requires accurate modeling of the long-term trajectory and short-term tendency of the object's movement. For example, an object in the video should move on the correct path (long-term) and exhibit natural and smooth actions (short-term) in a temporally-coherent video.

Given these challenges, current methods [13, 27, 36] for video generation have sought to extend the architecture of diffusion models from image generation. These methods typically employ a DDPM strategy and use U-Net models with 3D convolutions or temporal attention to capture and synthesize spatial-temporal motion features. To alleviate the limitations of 3D U-Net models, which are limited to generating fixed-length tablets with high memory consumption, [36] proposes an autoregressive approach that generates consecutive frames starting from an initial frame. However, autoregressive methods have limitations in their capacity to model long-term features, such as global motion trajectories, and tend to be sensitive to short-term motions. This can lead to the accumulation of errors or the generation of trivial, static movement in the synthesized videos. Meanwhile, previous methods [13, 27, 36] have mainly employed temporal convolution or attention to RGB space to model local temporal features within a fixed range. Our research has demonstrated that by enhancing local motion trend modeling through the adoption of cross-attention on explicit motion features, it is possible to generate more fluent videos.

In this paper, we propose a Motion-aware Video Diffusion Model (MV-Diffusion) for generating high-quality videos with coherent temporal structures. To achieve this, we focus on modeling both long-term motion trajectories and short-term motion trends. For **Trajectory Modeling**, we argue that it is crucial to combine global motion direction and local movement patterns in an appropriate manner. In our implementation, we use convolution kernels to extract global motion direction features from long-range optical flow and capture local movement features from adjacent past frames. These features are then combined as trajectory information to guide the network in synthesizing meaningful and realistic motions. To model the motion trend, we introduce a **Motion Trend Attention** mechanism based on cross-attention in the autoregressive architecture. We assume that the motion trend can be inferred from the motion intention in the near past. To that end, we estimate the optical flow between the first video frame and the adjacent past frames to model the object's motions. A cross-attention unit is used to learn the motion intention from these optical flow features, which is then incorporated into the U-Net network. This learned motion trend guides the generation of the current frames, resulting in a coherent and smooth video.

Our main contribution can be summarized as follows:

- We propose a **Trajectory Modeling** block to enhance the conditioning of the U-Net model with long-term trajectory features. This block effectively avoids trivial static movement in autoregressive approaches.
- We propose a **Motion Trend Attention** block that explicitly captures short-term motion features through the use of a cross-attention mechanism. This improves the smoothness and coherence of the generated videos.
- We conducted in-depth experimentation on three downstream tasks: video generation, prediction, and interpolation using four datasets: UCF101, BAIR, SM-MNIST, and KTH. We present both quantitative and qualitative evidence of the efficacy of our methods and perform ablation studies on our design methods.

2 RELATED WORK

The advancement of video generation techniques has been closely tied to the development of generative models.

2.1 Autoencoders and GAN-based Methods

Early studies on video generation [8, 22, 30] mainly combine convolution and recurrent neural networks to predict successive video frames. To increase the diversity of the synthesized videos, [1, 2, 5, 9] introduce stochastic latent variables to predict different possible futures.

After that, the development of Generative Adversarial Networks (GANs) significantly promotes the synthesis fidelity of video generation and prediction. MoCoGAN [32] proposes decomposing videos into motion and content to control both during generation. Further studies focus on generating high-resolution and long-duration videos. DIGAN [42] encodes videos into implicit neural representations to take video as continuous signals, which achieves generating videos with longer duration. StyleGAN-v [28] designs continuous motion representations to extend powerful image generation StyleGAN2 architecture to video generation, which further increases the resolution and quality.

Although these works show promising results in generating high-definition and long videos, these methods lack the ability to generate complex open-domain videos due to the inner drawback of GANs. In addition, limited by the locality bias of convolutional neural networks, the GAN-based methods struggle on complex scenes and multiple objects [11].

2.2 Transformer-based Methods

Transformer architecture is popularly applied in natural language processing and computer vision tasks. This architecture greatly benefits from model scaling up yet can only work on discrete tokens. With VQ-VAE[34] introducing a way of discretizing visual patches, Latent Video Transformer [19] and VideoGPT [41] propose to use transformer architecture that generates inner latent tokens in a spatial autoregressive manner. Nuwa [39] improves upon this by adopting a 3D transformer encoder-decoder framework. CogVideo [14] further proposes a multi-frame-rate training strategy to realize generating videos of flexible fps. Although these methods show great potential to generate open-domain videos via scaling up the model size, they suffer from unidirectional bias, accumulated prediction errors[11], and huge computational costs.

2.3 Diffusion Model-based Methods

Denoising Diffusion Probabilistic Models are first proposed in [12]. Inspired by nonequilibrium thermodynamics, it learns the distribution of the target data domain by gradually reconstructing the data. Specifically, it first maps the data to Gaussian distribution by gradually adding noise, and then learns a denoising network to reverse the noise-adding process. Dhaliwal et al. [6] improve the denoising U-Net architecture and apply the diffusion model to the image synthesis, which achieves better generative results than GANs for the first time. After that, research works [11, 17, 20, 24] further employ Denoising Diffusion Probabilistic Models to text-to-image generation and greatly advance the generative quality.

VDM [13] extends the 2D architecture to 3D by keeping its original components as spatial-only operations and employing additional temporal attention to capture temporal features. Make-a-video [27] and Phenaki [35] further apply video diffusion models to high-resolution and long-duration video generation. However, both of them rely on large models and consume vast amounts of computation resources. Moreover, they lack explicit constraints on the temporal coherence of the generated videos. MCVD [36] proposes to reduce the redundancy of the architecture by generating videos in a temporal autoregressive manner. Additionally, it uses conditional masks to unify video generation, prediction, and interpolation in a single architecture. Although efficient to generate videos, current autoregressive approaches lack the capability to

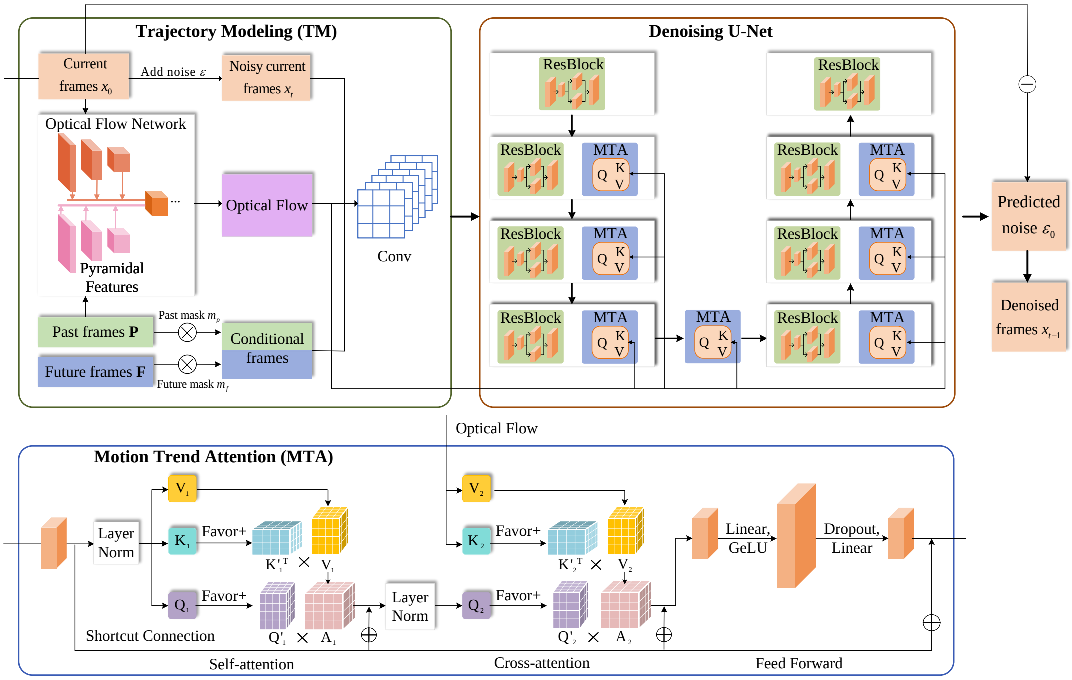

Figure 1: The framework of the Motion-aware Video Diffusion Model. The architecture utilizes an autoregressive approach, in which the denoising U-Net leverages noisy current frames at time T to predict current frames in the DDPM setting based on conditional frames. We design a Trajectory Modeling block and a Motion Trend Attention block to improve the temporal modeling.

sense global motion information to generate temporal coherent videos.

3 MOTION-AWARE VIDEO DIFFUSION MODEL

In this paper, we propose a video generation pipeline that utilizes an autoregressive approach [36], in which noisy frames at time T are used as the starting point to predict previous frames through a denoising U-Net in the DDPM setting. The pipeline is designed to improve temporal modeling by incorporating two core components: Motion Trend Attention (MTA) and Trajectory Modeling (TM), as illustrated in Figure 1. The Trajectory Modeling block extracts trajectory information based on the optical flow between past and current frames, while the Motion Trend Attention employs self-attention and cross-attention mechanisms to learn motion trends. In addition, we have adopted a lightweight design in the denoising U-Net architecture to reduce computation costs and enhance efficiency. We will provide a detailed description of our methods in the subsequent section.

3.1 Video Diffusion Methods

To provide a clear understanding of our approach, which is based on video diffusion, we begin by providing a brief overview of the underlying algorithm. Video diffusion is based on the Denoising Diffusion Possibility Model (DDPM) [12], which models the distribution of real data through a denoising process. The architecture is extended to videos in video diffusion. Specifically, a forward process is employed to map videos into Gaussian noise through the addition of noise in a Markov chain. Let x_0 be a sample video from the distribution $x_0 \sim q(x_0)$, where the length of the Markov chain is T. The noising adding process can be formulated as follows:

$$q_t\left(x_t \mid x_{t-1}\right) := \mathcal{N}\left(x_t; \sqrt{1-\alpha_t}x_{t-1}, \alpha_t \mathbf{I}\right), \tag{1}$$

where $t \in [1, T]$ is the timestep, x_t denotes the sample in step t. Moreover, the x_t can be directly derived by x_0:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{2}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}\left(1-\alpha_s\right)$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. As the forward process progresses, x_t can be considered as being equivalent to pure Gaussian noise.

With regards to the denoising process, the posterior distribution $q\left(x_{t-1} \mid x_t\right)$ can be approximated as a Gaussian distribution when the value of α_t is sufficiently small. The reverse conditional probability is computationally tractable when conditioned on x_0 through the application of Bayes' rule:

$$q_t\left(x_{t-1} \mid x_t, x_0\right) := \mathcal{N}\left(x_t; \hat{\mu}(x_{t-1}, x_0), \hat{\sigma}\right), \tag{3}$$

by integrating equation 2, we can derive that

$$\hat{\mu}(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right). \tag{4}$$

Therefore, to reverse the noise-adding process, we learn a denoising model θ that predicts the added noise ϵ at each timestep t. The training objective of the model θ can be simplified as follows:

$$L(\theta) = E_{t\sim[1,T], x_0\sim q(x), \epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\left\|\epsilon - \epsilon_\theta\left(x_t \mid t\right)\right\|^2\right] \tag{5}$$

3.2 Autoregressive Generation Approach

With model θ, successive denoising steps can be taken on pure Gaussian noise to generate a video. However, this model is limited in its ability to generate videos of variable length unconditionally. To address this limitation, [36] employs an autoregressive generation approach, which extends the video based on past frames. Given the first p frames, the model predicts n frames at a time to synthesize a longer video. To unify video prediction and video generation within a single framework, specifically, the generation of videos without the first p frames, a binary mask m_p is applied to the past frames $\mathbf{P}$.

Additionally, the framework is extended to the task of video interpolation through the application of a binary mask m_f to the future frames $\mathbf{F}$. The loss function $L_{unify}(\theta)$ can be formulated as follows:

$$\begin{aligned} L_{unify}(\theta) = E_{t\sim[1,T],[\mathbf{P},x_0,\mathbf{F}]\sim q(x),\epsilon\sim\mathcal{N}(0,\mathbf{I})} \\ [\|\epsilon - \epsilon_\theta(x_t \mid t, m_p\mathbf{P}, m_f\mathbf{F})\|^2] \end{aligned} \tag{6}$$

The binary masks m_f and m_p allow the model to synthesize videos without the need for past or future frames. As a result, the network trained on $L_{unify}(\theta)$ can accomplish three tasks: video generation, prediction, and interpolation.

While the Autoregressive model is efficient in generating videos, it lacks the capability to effectively capture global motion information. To address this limitation, we propose to enhance the model's conditioning through the incorporation of trajectory modeling. Additionally, while motion trends can be inferred implicitly from previous frames, we have found that explicitly enhancing temporal modeling through the use of cross-attention can lead to the generation of more coherent videos.

3.3 Trajectory Modeling

Our trajectory modeling aims to capture long-term motion trajectories from past video frames, which guides the denoising network to generate meaningful motion. We posit that the global directions and local patterns of motion of the object are critical in determining trajectories. To that end, we implicitly model motion trajectories by learning global directions from long-term optical flow and local movement patterns from adjacent video frames. Specifically, given the past frames $\mathbf{P} = p^1, p^2, ..., p^{n_p} \in R^{B\times n_p C\times H\times W}$ and future frames $\mathbf{F} = f^1, f^2, ..., f^{n_f} \in R^{B\times n_f C\times H\times W}$, binary masks are applied to them to allow the network to synthesize videos without conditional frames.

$$X_{cond} = f([m_p\mathbf{P}, m_f\mathbf{F}]) \tag{7}$$

where m_p, m_f refer to the binary mask, n_p, n_f are the number of the past frames and future frames, and $f([.])$ denotes concatenating operation. $B\times C\times H\times W$ denote batch size, channel size, frame height, and frame width, respectively. Note that we preprocess a video frame as 2D data to reduce computation costs. These conditional frames contain the local movement pattern information that we need to model the motion trajectory. The optical flow estimation network G_o extracts optical flow X_{flow} between the first frame x^1 and past frames P to establish global directions of the motion:

$$X_{flow} = f([G_o(x^1, p^1), G_o(x^1, p^2), ..., G_o(x^1, p^{n_p})]) \tag{8}$$

Then we use convolution kernels $\mathrm{K}_{\mathrm{traj}} \in R^{3\times3\times C_i\times C_o}$, where $c_i = 2n_p + C(n_p + n_f)$, to extract trajectory information based on the global direction and local pattern features:

$$X_{traj} = K_{traj} * f([x_t, X_{flow}, X_{cond}]) \tag{9}$$

where x_t is the noisy current frame that the denoising network input. The trajectory-enhanced features X_{traj} are then sent into the denoising U-Net to predict the added noise in the t step.

3.4 Motion Trend Attention

The Motion Trend Attention is designed to extract motion trends from optical flow features and guide the network to generate coherent videos. We posit that the current motion trend can be inferred by analyzing the motion intention in the recent past. Based on this principle, our Motion Trend Attention first employs self-attention to extract the current motions from features of the current frames. Subsequently, a cross-attention unit is utilized to incorporate the motion trend from the past by learning motion intention from the optical flow features of past frames. The resulting motion trend is then utilized to guide the generation of current frames, resulting in the formation of a smooth video.

Given the feature maps of the current frames X, we calculate the query, key, value Q_1, K_1, V_1 with three different linear layers:

$$Q_1 = \bar{X}W_{q_1} + b_{q_1}, K_1 = \bar{X}W_{k_1} + b_{k_1}, V_1 = \bar{X}W_{v_1} + b_{v_1} \tag{10}$$

where $W_{q_1}, W_{k_1}, W_{v_1}, b_{q_1}, b_{k_1}, b_{v_1}$ are the parameters of the linear layers, $\bar{X} = Norm(X)$. The linear layer here also functions as a projector, bridging the cross-modal gap between flow and features. To reduce computation costs, we use the FAVOR+ algorithm [4] to find Q_1', K_1' that satisfy:

$$X_1 = (Q_1K_1^\top/\sqrt{d})V_1 = Q_1'(K_1'^\top V_1) \tag{11}$$

The output of the self-attention X_1 is forwarded into the query of the cross-attention unit. The key and value are calculated with the optical flow feature X_{flow}:

$$\begin{aligned} Q_2 = \bar{X_1}W_{q_2} + b_{q_2}, K_2 = X_{flow}W_{k_2} + b_{k_2}, \\ V_2 = X_{flow}W_{v_2} + b_{v_2} \end{aligned} \tag{12}$$

where $W_{q_2}, W_{k_2}, W_{v_2}, b_{q_2}, b_{k_2}, b_{v_2}$ are the parameters of the linear layers, $\bar{X_1} = \mathrm{Norm}(X_1 + X)$. With the FAVOR+ algorithm, we can find Q_2', K_2' that satisfy:

$$X_2 = (Q_2K_2^\top/\sqrt{d})V_2 = Q_2'(K_2'^\top V_2) \tag{13}$$

The output of the cross-attention X_2 is sent into a feed-forward unit, which can be formulated by:

$$X_3 = \mathrm{Dropout}(g(\bar{X_2}W_{ff_1} + b_{ff_1}))W_{ff_2} + b_{ff_2} \tag{14}$$

where $W_{ff_1}, W_{ff_2}, b_{ff_1}, b_{ff_2}$ are the parameters of the linear layers to scale the feature maps, $\bar{X_2} = Norm(X_2 + X)$, $g(.)$ denotes the GeLU activation function. Finally, X_3 is added with the shortcut connection and forwarded into the next unit of the network.

3.5 Lightweight Design

In order to balance computational efficiency and performance, given that the Trajectory Modeling and Motion Trend Attention blocks increase the computation cost of the network (as observed in Table 6), we implement lightweight design techniques as shown in Figure 2.

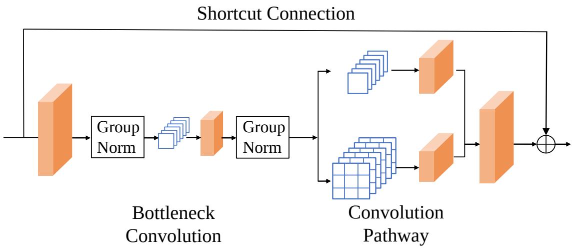

Figure 2: The lightweight network design we use to reduce the computation cost.

Given the input feature maps Y, we first normalize them and send them into a bottleneck pointwise convolution F_1 to reduce its channels. After conducting a group normalization, the feature maps are forwarded into two convolution pathways with pointwise convolution F_2 and 3×3 convolution F_3 respectively. Finally, we concatenate the output of two pathways. The process can be formulated as:

$$\mathrm{Y}_{\mathrm{out}} = f([\overline{\bar{Y} * F_1} * \mathrm{F}_2, \overline{\bar{Y} * F_1} * \mathrm{F}_3]) \tag{15}$$

where $\bar{Y} = \mathrm{GroupNorm}(Y)$, $*$ denotes convolution operation. The output of the convolution pathway, denoted as $\mathrm{Y}_{\mathrm{out}}$, is further added with the shortcut connection before being passed on to the next block. The bottleneck convolution, represented by F_1, reduces the input channels of the convolution pathways, and the use of multiple pathways reduces the output channels. These techniques effectively minimize the computation cost associated with the convolution operation.

4 EXPERIMENT

4.1 Datasets

To evaluate the performance of our Motion-aware Video Diffusion Model, we conduct experiments on four video datasets: UCF101 [29], 13,320 real-world human action videos collected from YouTube; BAIR robot pushing [7], 44,000 videos of a robot’s arm pushing toy objects on the top of a table; SM-MNIST [5], videos of handwritten digits moving stochastically based on 60,000 black-and-white digit images; and KTH [26], 600 human action videos of 6 categories.

4.2 Evaluation Metrics

In evaluating the performance of our model, we primarily rely on the FVD metric [33]. This metric utilizes an I3D model pre-trained on Kinetics-400 to extract features from videos and calculates the distribution distance between the features of synthesized videos and real videos. A lower FVD score indicates that the network is capable of generating videos of higher fidelity and coherency. We use 2,048 real and fake video clips for evaluating FVD. Additionally, we also report SSIM [37] and PSNR metrics to measure the performance in video prediction and interpolation tasks. SSIM measures the structural similarity between video frames, taking into account luminance, contrast, and structural similarity. A higher SSIM score indicates that the generated videos are more similar to the ground truth. PSNR measures the ratio of noise to signal in video frames, and a higher PSNR score indicates a higher-quality video that is less likely to corruption by noise.

Table 1: Unconditional generation results on UCF-101, generating videos of 16 frames. * denotes the model is trained on train+test split, otherwise the method uses only the train split for training.

Method	Publications	FVD ↓
MDP [43]	ICCV 2019	1277.0
TGANv2 [25]	IJCV 2020	1209.0
MoCoGAN-HD* [31]	ICLR 2021	700.0
StyleGAN-v [28]	CVPR 2022	1431.0
MCVD concat [36]	NeurIPS 2022	1228.3
MCVD spatin [36]	NeurIPS 2022	1143.0
DIGAN [42]	ICLR 2022	655.0
DIGAN* [42]	ICLR 2022	577.0
CogVideo [14]	ICLR 2023	626.0
MV-Diffusion	Ours	**492.6**

Table 2: Unconditional generation results on BAIR dataset. *f* refers to the frames that the model generated.

Method	Publications	f	FVD ↓
MCVD-spatin [36]	NeurIPS 2022	30	399.8
MCVD-concat [36]	NeurIPS 2022	30	348.2
MCVD-spatin [36]	NeurIPS 2022	16	267.8
MCVD-concat [36]	NeurIPS 2022	16	228.5
MV-Diffusion	Ours	30	92.8
MV-Diffusion	Ours	16	**56.8**

4.3 Visualization of Motion Trend Attention

In this section, we provide a visualization of the Motion Trend Attention. Upon observing the attention map, we can clearly observe that the attention weight is concentrated on the body parts where the movement happens. For instance, in the case of blowing candles, the attention primarily focuses on the hand and mouth regions. This visualization provides compelling evidence that our Motion Trend Attention is capable of capturing short-term tendencies and effectively guiding the model in generating smooth actions.

4.4 Implement Details

The Motion-aware Video Diffusion Model utilizes the DDPM algorithm during both the training and sampling stages. The model’s performance is evaluated through experimental results obtained by utilizing 100 DDPM steps for inference. To account for variations in dataset size, models of varying dimensions are employed, with the

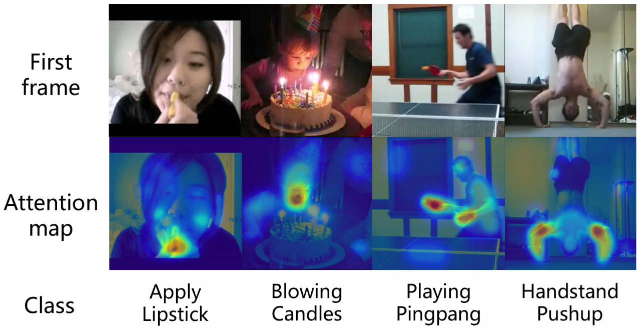

Figure 3: A visualization of the Motion Trend Attention

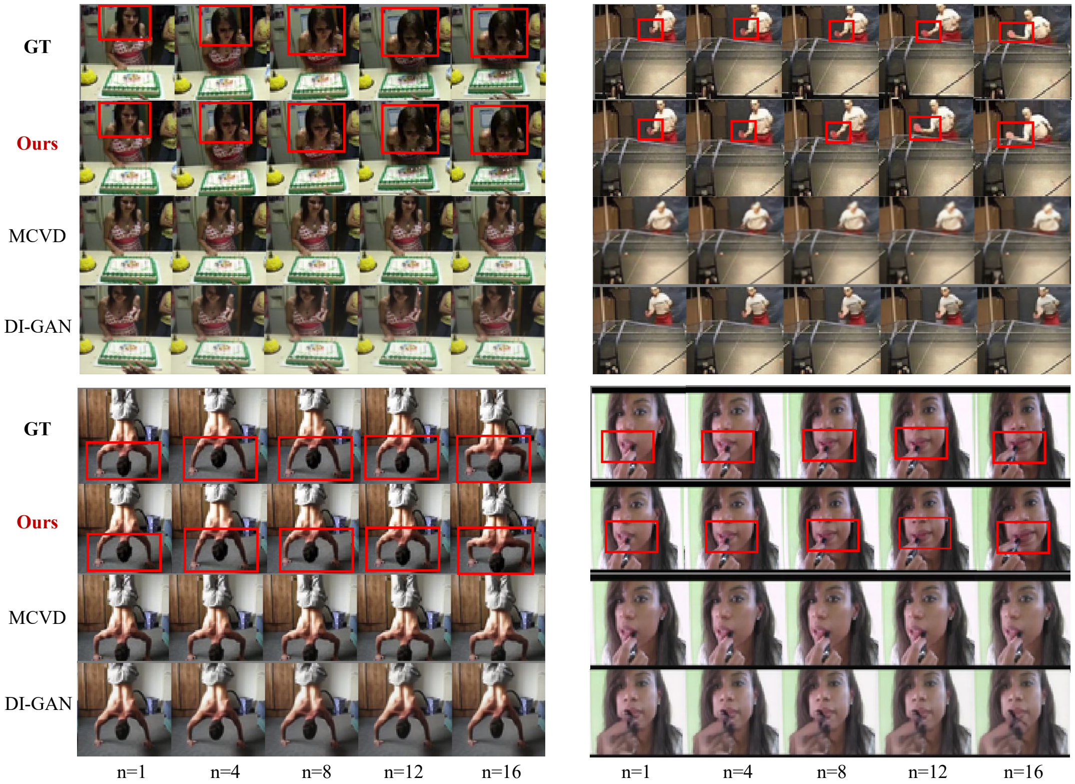

Figure 4: Comparison of generated videos between MV-Diffusion, MCVD, and DI-GAN on UCF. The corresponding real videos are also shown for reference denoted as "GT" in the first row. We use red rectangles to mark the actions in the videos.

same settings as those utilized in the MCVD [36]. We employ pretrained SpyNet [21] to estimate optical flow, maintaining prediction accuracy during early training stages, while also fine-tuning its parameters to enhance accuracy throughout the training process. Our model is engineered to generate motion patterns both with and without flow. This capability ensures the generation of motion patterns even in cases with weak or no flow. Furthermore, incorporating flow serves to enhance the generated motion patterns in subsequent frames. In the case of the UCF dataset, class labels are not utilized and the traditional train-test split is maintained. All video frames undergo center cropping and are resized to 64 × 64 pixels for both training and testing purposes. The model is trained on 1-4 Nvidia A40 or Tesla v100 GPUs.

4.5 Comparison with the State-of-the-art

To compare our Motion-aware Video Diffusion Model with existing methods, we test the performance of our model in video generation, prediction, and interpolation tasks and report the results in Table 1-5.

4.5.1 Video Generation. In this experiment, we compare the performance of our method with existing methods on the UCF and BAIR datasets. Table 1 presents the results on the UCF dataset, demonstrating that our approach outperforms all comparison methods. Despite the fact that CogVideo utilizes models with 9.4 billion parameters pre-trained on 5.4 million data, our approach still outperforms it while only employing models with 374 million parameters (4% of CogVideo) and without utilizing any extra data. The improved performance of our approach can be attributed to its elaborate modeling of motion, which allows for the generation of a small number of frames while maintaining video smoothness. In contrast, CogVideo utilizes large Transformer-based models to synthesize complete videos, resulting in redundancy and inefficiency. Additionally, CogVideo relies on textual input for simple motion information, while our approach extracts more concrete motion trajectory and tendency features. As a result, our approach is capable of generating more realistic motions and achieves a lower FVD.

Table 3: Video prediction results on SM-MNIST (64 × 64) for 10 predicted frames conditioned on 5 past frames.

Method	Publications	FVD ↓
SVG [5]	ICML 2018	90.81
vRNN 1L [3]	ICCV 2019	63.81
Hier-vRNN [3]	ICCV 2019	57.17
MCVD concat [36]	NeurIPS 2022	25.63
MCVD spatin [36]	NeurIPS 2022	23.86
MV-Diffusion	Ours	**16.59**

Additionally, We provide qualitative results on the UCF dataset in Figure 4, incorporating ground truth (GT) and comparisons to prior works. Our evaluation, featuring scenarios such as *blowing candles, playing ping pong, performing handstands, and applying lipstick*, reveals that MCVD and DI-GAN generate **static scenes**

Table 4: Video prediction results on BAIR robot pushing (64 × 64), predicting 15 frames based on one frame.

Method	Publications	FVD ↓	PSNR ↑	SSIM ↑
Video Transformer [38]	ICLR 2020	96.0	-	-
CCVS [15]	NeurIPS 2021	99.0	-	-
MCVD-concat [36]	NeurIPS 2022	98.8	18.8	**0.829**
MCVD-spatin [36]	NeurIPS 2022	103.8	18.8	0.826
MCVD concat pf-mask [36]	NeurIPS 2022	89.5	16.9	0.780
NVWA [39]	ECCV 2022	86.9	-	-
VDM [13]	ICLR 2022	66.9	-	-
MV-Diffusion concat pf-mask	Ours	64.5	**18.9**	**0.829**
MV-Diffusion concat	Ours	**54.6**	18.8	**0.829**

Table 5: Video interpolation results on SM-MNIST, KTH, and BAIR robot pushing datasets. Given 9 past frames and 9 future frames, the model interpolates 7 frames. MCVD uses fewer past frames and future frames. We follow the same setting as MCVD to fairly compare with it. * denotes only interpolates 5 frames.

Method	Publications	SMNIST		KTH		BAIR	
		PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
FSTN [16]	ICCV 2017	14.730	0.765	29.431	0.899	19.908	0.850
SepConv [18]	CVPR 2017	14.759	0.775	29.210	0.904	21.615	0.877
SVG-LP [5]	ICML 2018	13.543	0.741	28.131	0.883	18.648	0.846
SDVI full [40]	WACV 2020	16.025	0.842	29.190	0.901	21.432	0.880
SDVI [40]	WACV 2020	14.857	0.782	26.907	0.831	19.694	0.852
MCVD [36]	NeurIPS 2022	20.944	0.854	34.669	0.943	23.408	0.914
MCVD* [36]	NeurIPS 2022	27.693	0.941	35.611	0.963	25.162	0.932
MV-Diffusion	Ours	22.258	0.878	34.292	0.944	24.299	0.926
MV-Diffusion*	Ours	**29.841**	**0.966**	**37.448**	**0.968**	**26.678**	**0.948**

lacking meaningful motions. Conversely, our method demonstrates superior temporal consistency, excelling in two key aspects: **(1) Coherent actions**: The generated videos display smooth and natural behavior, closely resembling the motion patterns observed in the GT videos. This similarity is achieved through our motion trends attention (MTA) module, which effectively captures short-term motion features to model movement tendencies. **(2) Meaningful motions**: Motions are not only easily recognizable but also contextually relevant, as our trajectory modeling (TM) approach enhances the U-Net model’s conditioning with long-term trajectory features, preventing trivial motions.

In addition to the primary experiment, we also conduct experiments with varying frame settings on BAIR, as illustrated in Table 2. Our approach demonstrates a significant performance improvement, as evidenced by an FVD score that is only **1/4** of the previous state-of-the-art autoregressive method, MCVD. This improvement is attributed to the utilization of long-term optical flow to extract global motion directions and model the trajectory, as opposed to MCVD’s reliance on short-range autoregressive networks. This allows for the generation of more accurate object motions that closely resemble real videos, resulting in a lower FVD score.

4.5.2 Video Prediction. We evaluate the performance of our Motion-aware Video Diffusion Model in video generation tasks on the BAIR and SM-MNIST datasets. The results on the BAIR datasets, presented in Table 4, indicate that our model significantly outperforms the previous state-of-the-art VDM. This superior performance can be attributed to our approach’s ability to model motion trajectory and tendency, which VDM fails to capture as it only applies temporal attention blocks on raw RGB-based features and learns ambiguous global motion trends implicitly. On the other hand, our approach explicitly models the motion trajectory by learning both global motion directions and local movement patterns, and the motion trend attention extracts motion trends by discovering motion intention in the near past. As a result, our approach captures more accurate motion information and generates more realistic motions, resulting in a lower FVD score. Our approach also achieves the best performance on the SM-MNIST dataset, as shown in Table 3. Our FVD score is **30%** lower than the previous state-of-the-art MCVD, further validating the superiority of our approach to predict more accurate motions.

4.5.3 Video Interpolation. The performance of our approach in video interpolation tasks is evaluated on the SM-MNIST, KTH, and BAIR datasets. The results, presented in Table 5, indicate that our approach outperforms the previous state-of-the-art MCVD. The PSNR scores of our approach are 2.148, 1.837, and 1.516 higher than the previous best results on the three datasets respectively. Our approach also shows a clear improvement over MCVD on SSIM, despite the fact that it has already achieved promising results. This is notable as our approach primarily relies on motion information learned from the past, which is less useful in interpolation tasks as future frames are given. Nevertheless, the PSNR and SSIM scores of our approach are still higher, indicating a higher similarity to the ground truth videos and higher quality of the interpolated frames. our approach consistently demonstrates superior performance in these metrics across three distinct datasets. Our empirical analysis suggests that this improved performance is attributed to our approach’s ability to leverage the long-term motion trajectory and tendency learned from past frames to synthesize more realistic motions, resulting in more continuous interpolated frames.

Table 6: Ablation studies on unconditional video generation task, using BAIR dataset with 16 frames generated. LD refers to the Lightweight Design. MTA and TM stand for Motion Trend Attention and Trajectory Modeling Block respectively.

Method	LD	MTA	TM	Params	FLOPs	FVD ↓
Baseline (MCVD)	×	×	×	251.2M	83.3G	228.5
+LD	√	×	×	112.4M	19.1G	218.3
+LD+MTA	√	√	×	167.1M	33.5G	80.6
+LD+TM	√	×	√	113.8M	20.4G	67.5
MV-Diffusion (Ours)	√	√	√	167.1M	34.9G	**56.8**

4.6 Ablation Study

To evaluate the effectiveness of each component of our Motion-aware Video Diffusion Model, we conduct ablation studies on the BAIR dataset in the video generation task, as shown in Table 6. The baseline used in the experiment is the MCVD-concat model, with the same model settings used by MCVD in the BAIR dataset.

First, we add a lightweight design to the baseline, denoted as "+LD" in Table 6. The results show that this lightweight design reduces 55% of the parameters and 77% of the computation cost while maintaining the generation fidelity of the model. To further illustrate the effectiveness of our lightweight design, we provide inference time comparisons on BAIR and UCF datasets in Table 7. The results show the superiority of our model in speed.

Second, we add the Motion Trend Attention (MTA) component to the lightweight architecture, denoted as "+LD+MTA" in Table 6. The FVD metric drops from 218.3 to 80.6, indicating that the MTA component guides the model to generate coherent and smooth videos as a lower FVD means higher fidelity and coherency.

Third, we add the trajectory modeling (TM) block to the lightweight architecture, denoted as "+LD+TM" in the table. The FVD metric drops to less than 1/3 of its original value, indicating that the trajectory modeling block helps the model generate more realistic motions.

Finally, we present the result of our full model in the last line of the table. With all the components, the computation cost of our approach is still significantly lower than the baseline, and the FVD metric further drops compared to "+LD+TM" and "+LD+MTA", verifying the effectiveness of both the MTA and TM components. Note that the TM module includes an optical flow estimation network with an additional 1.4M parameters, as shown by the difference between "+LD" and "+LD+TM" in Table 6. Since TM and MTA share this flow estimation network, there is no parameter increase between "+LD+TM" and the final solution.

These results demonstrate that all the components of our model contribute to the performance, and combining them leads to the best performance.

4.7 generalization ability

To analyze the generalization ability of our model, we conducted experiments where our model predicted actions on images depicting animals engaging in specific actions. These examples were distinct from those in the train/test datasets.

To carry out these experiments, we utilized Stable Diffusion[23] to generate the initial frames and then deployed our model, which had been trained on the UCF dataset, to predict the subsequent frames. We have shown the results of these experiments in Figure 5.

Table 7: FLOPs and Inference time comparisons on BAIR and UCF datasets. The inference time results are tested on a single Tesla A40 GPU.

Dataset	Method	FLOPs	Latency
BAIR	MCVD	83.3G	43.2ms
BAIR	VideoGPT	188.3G	138.6ms
BAIR	VDM	171.1G	128.2ms
BAIR	**MV-Diffusion(Ours)**	**34.9G**	**28.5ms**
UCF	MCVD	185.2G	75.5ms
UCF	VideoGPT	183.6G	74.2ms
UCF	VDM	586.7G	275.0ms
UCF	**MV-Diffusion(Ours)**	**73.2G**	**36.7ms**

The presented results demonstrate that our model is capable of predicting actions on unseen examples, thus showcasing its generalization ability. This analysis further reinforces the robustness and effectiveness of our proposed approach.

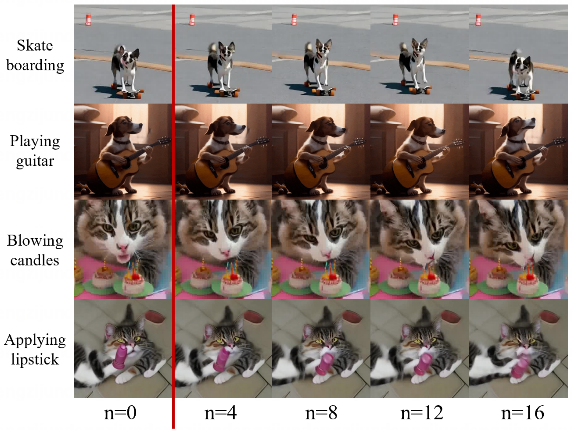

Figure 5: The generation results of animal actions for generalization ability analysis.

5 CONCLUSION

Generating temporally-coherent videos requires consideration of both the leveraging of global temporal information and the implementation of effective temporal modeling mechanisms. In this paper, we propose the Motion-aware Video Diffusion Model (MV-Diffusion), which leverages motion trajectory and tendency information learned from past frames. Our model includes a Trajectory Modeling block that learns global motion directions and local movement patterns to establish the motion trajectory. Additionally, the Motion Trend Attention component incorporates motion intention in the near past and improves the denoising U-Net layers through the use of attention mechanisms. Our method significantly improves upon the autoregressive video diffusion baseline and achieves state-of-the-art results on four video datasets and three video generation tasks.

6 ACKNOWLEDGEMENT

This work was supported by the grants from the National Natural Science Foundation of China (62132001, 61925201) and Kuaishou.

REFERENCES

[1] Adil Kaan Akan, Erkut Erdem, Aykut Erdem, and Fatma Güney. 2021. Slamp: Stochastic latent appearance and motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14728–14737.

[2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. 2018. Stochastic variational video prediction. In *International Conference on Learning Representations*.

[3] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. 2019. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7608–7617.

[4] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2021. Rethinking attention with performers. (2021).

[5] Emily Denton and Rob Fergus. 2018. Stochastic video generation with a learned prior. In *International conference on machine learning*. PMLR, 1174–1183.

[6] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.

[7] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. 2017. Self-Supervised Visual Planning with Temporal Skip Connections.. In *CoRL*. 344–356.

[8] Chelsea Finn, Ian Goodfellow, and Sergey Levine. 2016. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems* 29 (2016), 64–72.

[9] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. 2020. Stochastic latent residual video prediction. In *International Conference on Machine Learning*. PMLR, 3233–3246.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[11] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2022. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10696–10706.

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

[13] Jonathan Ho, Tim Salimans, Alexey A Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. 2022. Video Diffusion Models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*.

[14] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. 2023. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. (2023).

[15] Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. 2021. Ccvs: Context-aware controllable video synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 14042–14055.

[16] Chaochao Lu, Michael Hirsch, and Bernhard Scholkopf. 2017. Flexible spatio-temporal networks for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6523–6531.

[17] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *International Conference on Machine Learning*. PMLR, 16784–16804.

[18] Simon Niklaus, Long Mai, and Feng Liu. 2017. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 670–679.

[19] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. 2020. Latent video transformer. *arXiv preprint arXiv:2006.10704* (2020).

[20] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

[21] Anurag Ranjan and Michael J Black. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 4161–4170.

[22] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. 2014. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604* (2014).

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

[24] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487* (2022).

[25] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. 2020. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision* 128, 10 (2020), 2586–2606.

[26] Christian Schuldt, Ivan Laptev, and Barbara Caputo. 2004. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 3. IEEE, 32–36.

[27] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. 2022. Make-A-Video: Text-to-Video Generation without Text-Video Data. *arXiv preprint arXiv:2209.14792* (2022).

[28] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. 2022. 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*. 3626–3636.

[29] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[30] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*. PMLR, 843–852.

[31] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. 2021. A Good Image Generator Is What You Need for High-Resolution Video Synthesis. In *International Conference on Learning Representations*.

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

[33] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. 2018. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717* (2018).

[34] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017), 6306–6315.

[35] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. 2022. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399* (2022).

[36] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. 2022. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems* 35 (2022), 23371–23385.

[37] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[38] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. 2020. Scaling Autoregressive Video Models. In *International Conference on Learning Representations*.

[39] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. 2022. Nüwa: Visual synthesis pre-training for neural visual world creation. In *European Conference on Computer Vision*. Springer, 720–736.

[40] Qiangeng Xu, Hanwang Zhang, Weiyue Wang, Peter Belhumeur, and Ulrich Neumann. 2020. Stochastic dynamics for video infilling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2714–2723.

[41] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157* (2021).

[42] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. 2022. Generating Videos with Dynamics-aware Implicit Generative Adversarial Networks. In *International Conference on Learning Representations*.

[43] Vladyslav Yushchenko, Nikita Araslanov, and Stefan Roth. 2019. Markov decision process for video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 0–0.