

LaMD: Latent Motion Diffusion for Video Generation

Yaosi Hu¹ Zhenzhong Chen¹ Chong Luo²
 Wuhan University¹ Microsoft Research Asia²

Abstract

Generating coherent and natural movement is the key challenge in video generation. This research proposes to condense video generation into a problem of motion generation, to improve the expressiveness of motion and make video generation more manageable. This can be achieved by breaking down the video generation process into latent motion generation and video reconstruction. We present a latent motion diffusion (LaMD) framework, which consists of a motion-decomposed video autoencoder and a diffusion-based motion generator, to implement this idea. Through careful design, the motion-decomposed video autoencoder can compress patterns in movement into a concise latent motion representation. Meanwhile, the diffusion-based motion generator is able to efficiently generate realistic motion on a continuous latent space under multi-modal conditions, at a cost that is similar to that of image diffusion models. Results show that LaMD generates high-quality videos with a wide range of motions, from stochastic dynamics to highly controllable movements. It achieves new state-of-the-art performance on benchmark datasets, including BAIR, Landscape and CATER-GENs, for Image-to-Video (I2V) and Text-Image-to-Video (TI2V) generation. The source code of LaMD will be made available soon.

1. Introduction

Video generation aims to generate natural, high-quality videos that accurately reflect human intentions. Despite ongoing efforts, the field has yet to fully achieve this goal. One existing and straightforward solution to the video generation problem is to train a deep model which directly generates the pixel values for each frame in a video [11, 25, 29]. However, the sheer volume of video data makes it challenging to train the model effectively, and this lack of effectiveness has two implications. First, the training procedure requires significant computational resources, which can hinder the scalability of the model. Second, it may cause the generator to focus too much on spatial high-frequency details that are less noticeable to the human visual system. Therefore, developing a novel video generation framework

that can efficiently and effectively generate videos with a focus on perceptually relevant information is essential.

As an alternative to the above pixel-space generation approach, latent-space generation has become a preferred choice for reducing data redundancy and conserving resources during model training. This is achieved by using a pre-trained autoencoder to transfer the generation process from the pixel domain to a more efficient and compressed latent domain. Current latent-space video generation follows two typical paradigms, depending on the correspondence of the latent space. The first paradigm, as depicted in Fig. 1(a), employs an image autoencoder to transform each frame into latent tokens, effectively simplifying the video generation problem into an image generation problem in the latent space [19]. The second paradigm, as depicted in Fig. 1(b), employs a 3D video autoencoder to translate video clips into a latent space, and then generates content within this latent video space [55].

These two paradigms for latent-space video generation each have their own strengths and weaknesses. The first paradigm leverages recent advancements in latent-space image generation [16, 35, 38] and is capable of producing high-quality, semantically accurate frames. However, since 2D image autoencoders do not consider the temporal relationships between frames, the generated collection of frames may be temporally incoherent and not accurately depict motion. Conversely, the second paradigm implements spatiotemporal compression from video clips to the latent video domain, thereby overcoming the issue of motion coherence [21, 41, 59]. Nevertheless, modeling the video latent space, which requires 3D spatiotemporal information, presents significant challenges.

To address this challenge, it is crucial to focus on modeling the most perceptually relevant information in the video. We have observed that the inferior performance in video generation compared to image generation is due to the difficulty in producing consistent and natural motion in videos. By separating the visual appearance and motion components of a video, we can concentrate on motion generation specifically. This idea underpins our design for a new paradigm to address the generation problem in latent motion space, as depicted in Fig. 1(c). The use of a more compact

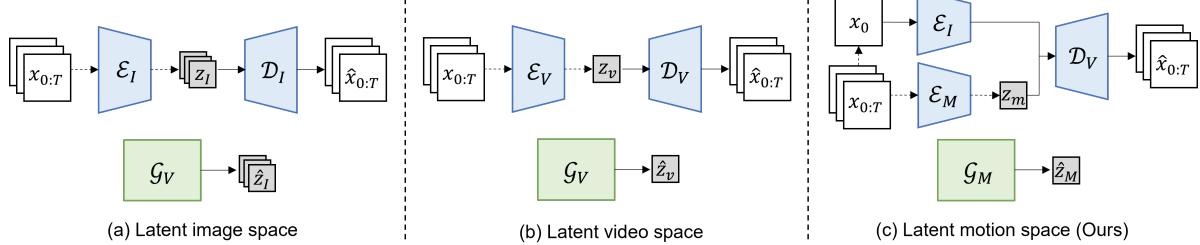


Figure 1. The comparison of video generation in different latent space. The dashed line stands for operations only involved in training process, while the solid line represents operations both involved in training and sampling process.

latent motion space instead of the latent video space reduces modeling complexity. This approach directly targets Image-to-Video (I2V) and Text-Image-to-Video (TI2V) generation, and with added Text-to-Image (T2I) generation, it can also solve the Text-to-Video (T2V) generation problem.

In this paper, we present the Latent Motion Diffusion (LaMD) framework, which represents a practical implementation of the Latent Motion Generation paradigm. The LaMD framework consists of two components, namely the Motion-Content Decomposed Video Autoencoder (MCD-VAE) and a diffusion-based motion generator (DMG). The MCD-VAE component tackles the challenge of separating motion from appearance or content by utilizing a multi-scale image encoder, a lightweight motion encoder with an information bottleneck, and a decoder for video reconstruction. The DMG module is influenced by the highly effective generative performance of Diffusion Models (DMs) [10, 24]. It gradually denoises motion variables over a continuous latent space and can also incorporate pre-obtained content features, as well as optional text, as multi-modal conditions. The computational complexity of the DMG module is comparable to that of image diffusion models, resulting in reduced resource utilization and faster sampling speed compared to other video diffusion models.

Our proposed LaMD framework has been evaluated on I2V and TI2V tasks using four commonly used datasets. The results show that the MCD-VAE achieves high reconstruction quality, with excellent compression of motion, and the DMG generates natural motion for video. LaMD also achieves new state-of-the-art performance on a variety of benchmarks, including BAIR, Landscape, and CATER-GENs which cover a wide range of motions from stochastic dynamics to highly controllable movements. In conclusion, our work makes the following contributions:

- We have introduced Latent Motion Generation as a new paradigm for video generation. The new paradigm focuses on the expressiveness of motion and therefore has the potential to generate videos with high perceptual quality.
- We have established a practical implementation of this

paradigm in the form of the Latent Motion Diffusion (LaMD) framework. The LaMD framework, which is made up of MCD-VAE and DMG, can produce videos with coherent and realistic motion under the appearance constraint. The option to incorporate text as a condition further increases control over motion.

- We have demonstrated the state-of-the-art capabilities of the LaMD framework through its results on four benchmark datasets for I2V and TI2V generation tasks.

2. Related Work

Video Generative Models have been developed in various types, including Variational Autoencoders (VAEs) [27], Generative Adversarial Networks (GANs) [18], autoregressive models (ARMs) [7], and Diffusion Models (DMs) [24]. These models have demonstrated remarkable success in generating high-quality videos, but each has its own limitations. VAEs attempt to learn latent variables z close to a prior distribution P_z by parameterizing the data distribution with a surrogate loss, but they may suffer from posterior collapse. While VAEs enable efficient sampling, the synthesized results are often blurrier than those of GANs, which are likelihood-free and based on the contest between two networks. GANs can achieve high-resolution synthesis with good perceptual quality but are unstable for training. Recently, DMs have shown promising results in various tasks which gradually inject and remove noise from data, but their sampling speed is much slower, as is that of ARMs. These generative models are earlier applied to generate videos in pixel domain. Recently, some work prefer to use a powerful autoencoder to transfer the generative goal from pixel space to latent space, which alleviates the burden of generative models and improves the sample quality.

Pixel Space Video Generation employs generative models to generate videos at the pixel level. Most of the existing methods focus on modeling the global motion representation from videos, i.e. by approaching a prior normal distribution to enable stochastic generation of videos from randomly sampled motion during inference

[2,11,45,51,57]. Some methods incorporate explicit actions into motion representation to support controllable generation [4, 31]. In Addition, DMs are recently used to model video space [25]. However, due to the high computational complexity of video data, these approaches are challenging for high-resolution video generation unless a spatial-temporal super-resolution module is added to generative models [23, 43]. Moreover, image compression techniques have shown that in general more than 80% of the data could be removed without noticeable decrease of perceptual quality [50]. But pixel-level generative models spend most of capacity to generate these imperceptible information, resulting in inefficient training. As a result, more research is focused on latent space video generation approaches.

Latent Space Video Generation typically adopts an autoencoder to handle the compression and reconstruction between pixel space and latent space, with subsequent generative model focused solely on latent space. Common image autoencoders like VQ-VAES [37, 47] and VQ-GANs [17, 56] achieve high-quality synthesis which compress image into discretized latent tokens. [58] has further modified it by modulating the quantized vectors to integrate spatially variant information, while [38] replaces the quantization layer with a slight KL-penalty over continuous latent variables. These powerful image autoencoders are widely used in image generation [6, 28, 34, 36], further directly incorporated into video generation by combining them with various generative models that operate on latent image space [20, 40, 42, 49, 53]. Since using image autoencoder to encode video into frame-wise latents can break the pixel-dependency across frames, [21, 48, 55] extend encoder from 2D-CNN to 3D-CNN, supporting video generation on latent video space. However, generating latent content and motion information simultaneously can be a challenging goal which would result in inefficient training.

The evolution of video generation from pixel space to latent space aims at moving the function of latent-to-pixel synthesis to a pre-trained autoencoder, and thereby allows generative model to focus on latent video generation instead. In this paper, we further transfer the motion-content synthesis to the autoencoder, thus allowing generative models to solely generate motion. Our approach is tailored for image-driven video generation, such as I2V and TI2V tasks, but can also be combined with image generation models to tackle other non-image-driven video generation tasks.

3. Method

The essence of motion lies in the dynamic movements across frames, such as trajectories. These movements are highly redundant in both spatial and temporal dimensions and can be naturally separated from content (i.e., appearance). Thus, to support video generation in latent motion space, we first design a motion-content decomposed video

autoencoder (MCD-VAE) which separates the representation of content and motion and further fuses them to reconstruct videos. Then a proposed diffusion-based motion generator (DMG) targets to generate content-specific motion latents conditioned on the pre-obtained content features with other optional conditions like text.

Our proposed two-stage approach LaMD is illustrated as Fig. 2. During training, we first train a MCD-VAE using video-only data and fix its parameters, and we then train the DMG in the latent motion space. During sampling, the DMG generates motion latents progressively from the normal distribution, guided by content feature extracted from the given image and other optional conditions. These motion latents are then fed into the MCD-VAE decoder to synthesize a video.

3.1. Motion-Content Decomposed Video Autoencoder

Targeting image-driven video generation, the motion component can be significantly compressed due to its high redundancy on both spatial and temporal dimension, while preserving more content information is beneficial to improve the quality of reconstruction. Thus, we employ two separate encoders to extract motion and content features, each with a different compression strategy.

Our MCD-VAE consists of the 2D-CNN based image encoder \mathcal{E}_I , the 3D-CNN based motion encoder \mathcal{E}_M , the fusion decoder \mathcal{D}_V . Given a video $x_{0:L} \in \mathbb{R}^{L \times H \times W \times 3}$ that contains L frames, the image encoder \mathcal{E}_I based on 2D UNet architecture [39] first encodes the first frame x_0 into content latents $f_{x_0} \in \mathbb{R}^{h \times w \times d'}$, as well as intermediate multi-scale representations $f_{x_0}^1, \dots, f_{x_0}^k$ through $\{f_{x_0}, f_{x_0}^1, \dots, f_{x_0}^k\} = \mathcal{E}_I(x_0)$, where k represents the number of scales. Meanwhile, the motion encoder \mathcal{E}_M extracts the motion latents from video based on 3D UNet architecture [8] with respective spatial and temporal down-sampling factor f_s, f_t . To achieve the decomposition of motion representation, an additional constraint is required on the motion branch. Here we employ an information bottleneck [1] that imposes a constraint on the KL divergence between the distribution of latent motion and a normal distribution. By controlling the size of bottleneck with a penalty corresponding to this KL divergence, the motion encoder can squeeze content information out and achieve motion decomposition. This process is parameterized as

$$\mathbf{z}_m = \mu_\theta(\mathcal{E}_M(x_{0:T})) + \varepsilon \cdot \sigma_\theta(\mathcal{E}_M(x_{0:L})) \quad (1)$$

where $\varepsilon \sim N(\mathbf{0}, \mathbf{I})$. Furthermore, considering a more compact manifold of the latent space is beneficial for the modeling of subsequent generator, the obtained latent motion z_m is then passed through a normalization layer, which we have found to effectively improve the stability and efficiency of training the diffusion-based generative model. Additionally,

Stage-I: Motion-Content Decomposed Video Autoencoder (MCD-VAE)

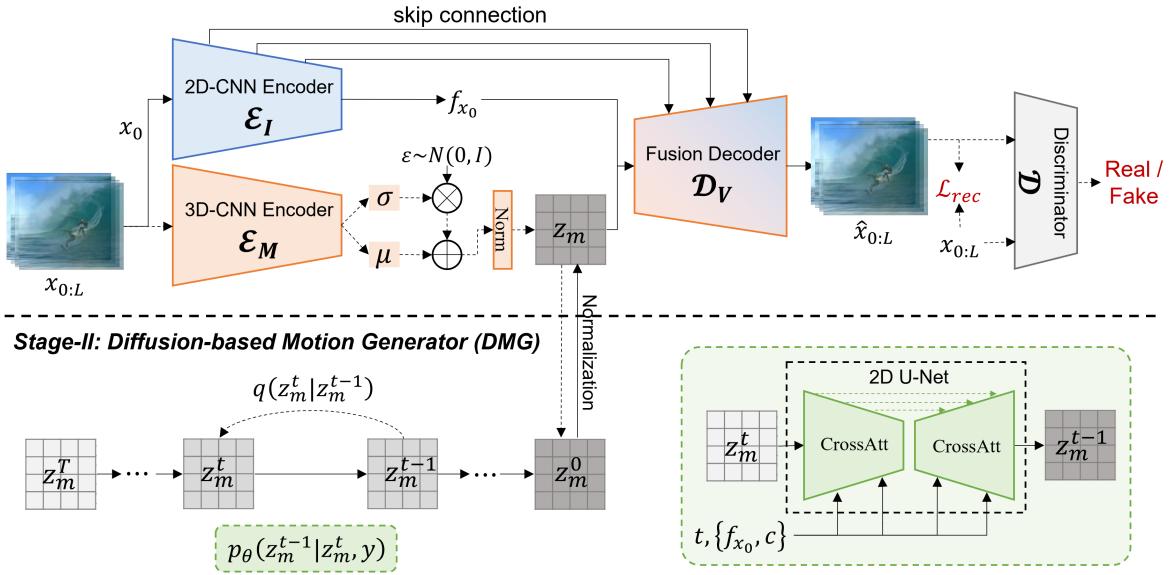


Figure 2. The framework of our proposed LMD. During training process, the stage-I MCD-VAE is first trained to decompose latent motion with video reconstruction task, while DMG is trained to generate natural motion conditioned by $\{f_{x_0}, c\}$ in the second stage. During sampling process, the motion latents are first generated by DMG and then input into the decoder \mathcal{D}_V together with multi-scale content features from the first given image to synthesize videos. The black dashed lines stand for operations only involved in training process.

it is worth noting that we set $f_t = L$ which means that temporal redundancy is fully discarded. Therefore, the latent motion representation is quite compact and of low dimensionality with $z_m \in \mathbb{R}^{h \times w \times d}$ where $h = \frac{H}{f_s}$, $w = \frac{W}{f_s}$.

The decoder aims at fusing content and motion to reconstruct video pixels. We modify the 3D UNet decoder by first fusing the spatial-aligned motion latents with deepest content feature, and then gradually incorporating multi-scale content features through skip connection. During the fusion, the spatial and temporal resolutions gradually increase until synthesizing the whole video $\hat{x}_{0:L} = \mathcal{D}_V(z_m, f_{x_0}, f_{x_0}^1, \dots, f_{x_0}^k)$.

The objective loss function contains the reconstruction loss which combines the L_1 pixel-level distance and LPIPS perceptual similarity loss [12], and the KL divergence with hyper-parameter β to control a clean decomposition of motion latents. Besides, based on [16], we also apply an adversarial objective via a video discriminator \mathcal{D} to improve the realism of reconstructions. Thus, the overall objective is formulated as

$$\begin{aligned} & \underset{\mathcal{E}_I, \mathcal{E}_M, \mathcal{D}_V}{\operatorname{argmin}} \max_{\mathcal{D}} \mathbb{E}_{x \sim p(x)} [\mathcal{L}_{GEN} + \lambda \mathcal{L}_{GAN}], \\ & \mathcal{L}_{GAN} = \log \mathcal{D}(x_{0:L}) + \log (1 - \mathcal{D}(\hat{x}_{0:L})), \\ & \mathcal{L}_{GEN} = \|x_{0:L} - \hat{x}_{0:L}\|_1 + \text{LPIPS}(x_{0:L}, \hat{x}_{0:L}) \\ & \quad + \beta \text{KL}(q_{\mu_\theta, \sigma_\theta, \mathcal{E}_M}(z_m | x_{0:L}) \| N(\mathbf{0}, \mathbf{I})), \end{aligned} \quad (2)$$

where λ stands for the adaptive weight in [16].

3.2. Diffusion-based Motion Generator

Based on the normalized motion on continuous latent space, we apply a diffusion-based motion generator (DMG) to learn the motion distribution $p(\mathbf{z}_m)$ via the reverse process of a fixed Markov Chain.

Following DMs [24, 44], our DMG consists of a forward diffusion process and a reverse denoising process. Given a latent motion $\mathbf{z}_m^0 \sim p(\mathbf{z}_m)$, the forward process is gradually adding Gaussian noises to \mathbf{z}_m^0 according to a series of scheduled variances β_1, \dots, β_T , producing a series of motion with increasing levels of noise $\mathbf{z}_m^1, \dots, \mathbf{z}_m^T$ as

$$q(\mathbf{z}_m^t | \mathbf{z}_m^0) := \mathcal{N}(\mathbf{z}_m^t; \sqrt{\bar{\alpha}_t} \mathbf{z}_m^0, (1 - \bar{\alpha}_t) \mathbf{I}), \quad (3)$$

$$\text{where } \bar{\alpha}_t = \prod_{s=1}^t \alpha_s, \alpha_t = 1 - \beta_t.$$

To recover \mathbf{z}_m^0 from \mathbf{z}_m^t , the reverse process progressively denoises via the parameterized Gaussian transition

$$p_\theta(\mathbf{z}_m^{t-1} | \mathbf{z}_m^t, y) = \mathcal{N}(\mathbf{z}_m^{t-1}; \mu_\theta(\mathbf{z}_m^t, t, y), \sigma_t^2 \mathbf{I}), \quad (4)$$

$$\mu_\theta(\mathbf{z}_m^t, t, y) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_m^t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{z}_m^t, t, y) \right), \quad (5)$$

where ϵ_θ is a trainable autoencoder to approximate the noise. y represents the conditions guiding the generating of motion, which include the pre-obtained content feature f_{x_0} to ensure realism of generated motion, as well as optional text inputs to provide additional control over the motion generation process. We follow [38] using a 2D UNet

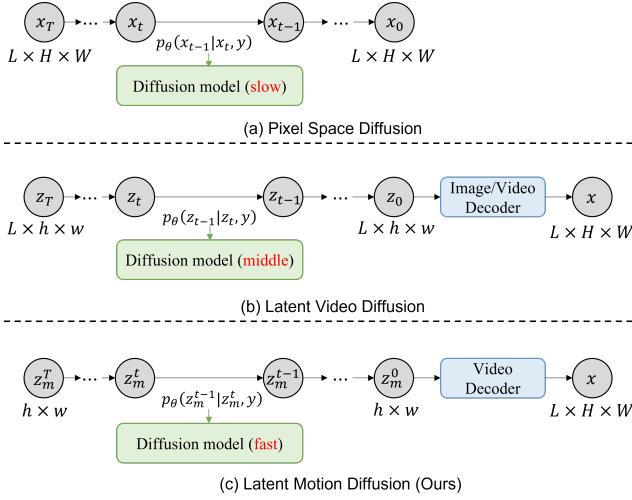


Figure 3. The comparison of sampling process of different video diffusion models. Benefited from low-dimensional diffusion target and 2D-CNN based diffusion model, our latent motion diffusion achieves much faster sampling speed compared to video space diffusion and latent video diffusion. The channel dimension is omitted in all settings for simplicity.

backbone as ϵ_θ with the cross-attention mechanism to incorporate conditions. The training objective can be simplified to

$$\mathcal{L}_{\text{simple}}(\theta) := \|\epsilon_\theta(\mathbf{z}_m^t, t, y) - \epsilon\|_2^2. \quad (6)$$

During sampling, we adopt the noise schedule from a subsequence with K sampling steps from $[1, 2, \dots, T]$ to improve the sampling speed [32]. Even though, in diffusion models, the generator ϵ_θ is recursively executed during sampling. As a result, the computational complexity of ϵ_θ can significantly impact the sampling speed, particularly for video data. Therefore, we analyze the sampling efficiency of latent motion diffusion by comparing the computational cost of different video diffusion models as depicted in Fig.3. Directly generating video pixels with dimension $L \times H \times W$ using pixel space diffusion based on a 3D UNet model results in the slowest sampling speed. In comparison, by compressing video to latent image/video space with a spatial downsampling ratio f_s and temporal downsampling ratio f_t ($1 \leq f_t < L$), latent video diffusion could reduce the computational complexity by $f_t \cdot f_s^2$ times under the same model architecture and channel size. In contrast, the complexity of our proposed latent motion diffusion, based on a 2D UNet structure, is at least $L \cdot f_s^2$ times less than pixel space diffusion, and even more when considering the temporal convolution in 3D UNet. Our latent motion diffusion achieves a computational cost comparable to image diffusion models and the fastest sampling speed among video diffusion paradigms.

4. Experiments

We evaluate our method on different image-driven video generation tasks: image-to-video and text-image-to-video. The experiments are conducted on four video datasets that cover a range of stochastic dynamics and highly controllable motion.

4.1. Datasets and Evaluation Metrics

Datasets: Here we summarize the datasets used for evaluation, which include Landscape and BAIR Robot Pushing for the I2V task, and CATER-GENs for the TI2V task.

- **BAIR Robot Pushing** [14] consists of videos about a robotic arm randomly pushing a variety of objects in a tabletop setting for real-world interactive agents. It contains about 40k training videos and 256 testing videos. Following the standard protocol [3, 33, 52], our model generates video with 16 frames given the first image at a resolution of 64×64 . We generate 100 samples conditioned per image and compare the 100×256 samples against 256 testing videos.
- **Landscape** [54] contains about 5k time-lapse videos which have been manually cut into short clips. Those scenes contain a wide range of content, weather, and motion speed, such as cloudy sky with moving clouds, and the starry sky with moving stars. This dataset contains 35,392 clips for training and 2,815 clips for testing. Following previous work [11, 54, 57], our model generates videos with 32 frames at a resolution of 128×128 for fair comparison.
- **CATER-GENs** [26] are synthetic datasets built in a 3D environment with realistic lighting and shadows. The movements of the objects are specified by detailed text descriptions. There are two versions of CATER-GENs, each with a different number of objects and attributes. CATER-GEN-v1 consists of two objects and four actions “rotate”, “contain”, “pick-place” and “slide”, including 3.5k training videos and 1.5k testing videos. CATER-GEN-v2 contains 3~8 objects from a large combination set of 4 kinds of attributes, including 24k training videos and 6k testing videos. We generate videos at a resolution of 128×128 and with a sequence length of 16 frames.

Implementation Details: The MCD-VAE downsamples the spatial dimension of latent motion by a factor of $f_s = 4$ and the temporal dimension by a factor of $f_t = \{16, 32\}$. The DMG applies similar 2D UNet backbone with cross-attention based conditional blocks as [38]. The MCD-VAE and DMG are both trained on the same training set of corresponding dataset without the use of external data, although the expressiveness and reconstruction quality of MCD-VAE

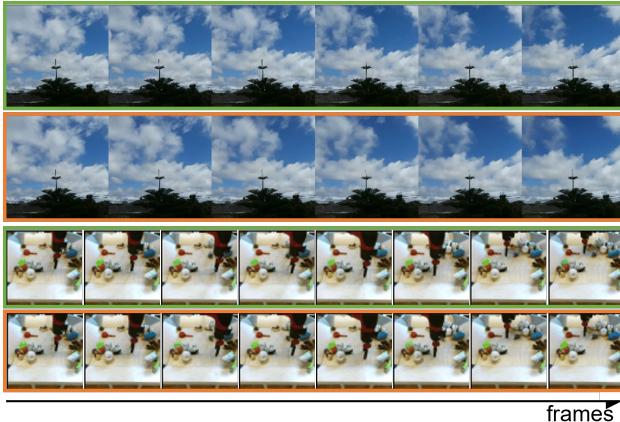


Figure 4. The reconstruction results of MCD-VAE on Landscape and BAIR Robot Pushing datasets. The original videos are contained in green boxes, while reconstructed videos in orange boxes.

Table 1. The quantitative results of reconstruction performance of MCD-VAE on BAIR Robot Pushing and Landscape datasets.

Datasets	f_s	f_t	d	FID \downarrow	LPIPS \downarrow	FVD \downarrow
BAIR	4	16	3	5.52	0.02	36.05
Landscape	4	32	4	2.11	0.09	37.35

could be further improved benefiting from extensive video data. During sampling, the latent motion is generated with a subsequence of $K = 200$ timesteps. More training details, architecture hyperparameters, and compute resources are listed in appendix.

Evaluation Metrics: Following previous work [11, 33, 55], we use the image-level perceptual similarity metrics including Fréchet Inception Distance (FID) [22] and Learned Perceptual Image Patch Similarity (LPIPS) [13], as well as the video-level metric Fréchet-Video-Distance (FVD) [46].

4.2. Reconstruction Ability of MCD-VAE

MCD-VAE is designed to perform motion decomposition and compression, as well as motion-content fusion and pixel-level reconstruction. Therefore, here we evaluate the decomposition and reconstruction ability of MCD-VAE through quantitative and qualitative analysis.

To evaluate the quality of reconstruction, we compare real videos with reconstructed videos which are synthesized from appearance feature of the first frame and latent motion of the video. Fig. 4 and Tab. 1 present the respective qualitative and quantitative results on the testing set of Landscape and BAIR Robot Pushing datasets. The results in Fig. 4 show that the motion latents successfully capture fine-grained movements such as the cloud deformation under a large motion compression ratio ($f_t = 32 \& f_s = 4$). The reconstructed videos also achieve high performance on both video-level and frame-level metrics in Tab. 1, reflecting that



Figure 5. The reconstructed video (bottom row) based on the appearance feature from the top row and the latent motion from the middle row.

Table 2. Ablation study of motion capacity in MCD-VAE on BAIR Robot Pushing datasets.

d	PSNR \uparrow	SSIM \uparrow	FID \downarrow	LPIPS \downarrow	FVD \downarrow
1	29.88	0.958	6.24	0.03	41.91
2	29.73	0.955	6.26	0.03	42.96
3	31.64	0.969	5.52	0.02	36.05
4	31.32	0.967	5.81	0.02	38.84

motion could be decomposed and highly compressed while still expressive enough to recover the video.

Because it is difficult to measure decomposition performance quantitatively, we assess the ability to decompose by transferring latent motion between videos. As shown in Fig. 5, the last video is synthesized from appearance features f of the first video and latent motion z_m of the second video. The movements in the synthesized video are consistent with the second video, while the appearance matches the first video. This confirms that MCD-VAE successfully decomposes motion without leaking appearance information.

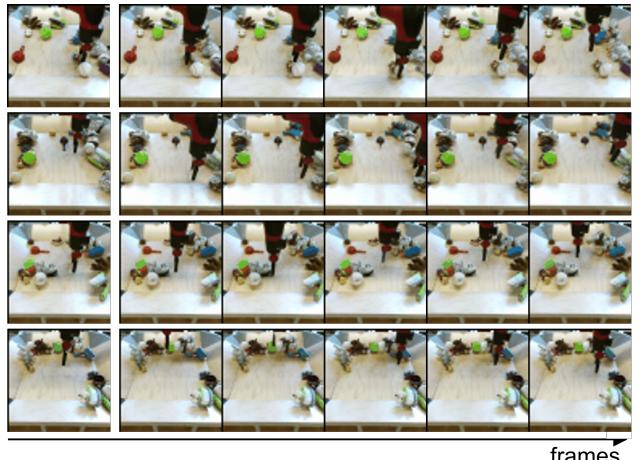


Figure 6. The generated samples on BAIR Robot Pushing dataset. Better viewed as video provided in the supplementary.



Figure 7. The generated samples on Landscape dataset. Better viewed as video provided in the supplementary.

We further discuss how much capacity is necessary for latent motion by changing its channel size d . We compare the reconstruction performance on different values of d in Tab. 2 (noted that the β to control the bottleneck size remains unchanged). The results indicate that MCD-VAE can achieve high reconstruction performance even with a small channel size, suggesting that there is significant redundancy in motion. Moreover, the findings suggest that video compression through motion decomposition and compression holds great potential.

4.3. Image-to-Video Generation

We first evaluate the performance of LaMD on I2V generation task which generates videos with one given image and stochastic dynamic movements.

For BAIR, we generate videos with 16 frames at a resolution of 64×64 as previous work. Generated samples in Fig. 6 shows that our model can generate natural movements.

Table 3. Quantitative evaluation compared to the state-of-the-art on BAIR.

Method	FVD ↓
LVT [33]	125.8
IVRNN [5]	121.3
DVD-GAN [9]	109.8
VideoGPT [55]	103.3
cINN [11]	99.3
Video Transformer [52]	94.0
FitVid [3]	93.6
NÜWA [53]	86.9
VDM [25]	66.9
LaMD (Ours)	57.0

Table 4. Quantitative evaluation compared to the state-of-the-art on Landscape. The results of methods for comparison are quoted from corresponding paper or reported by [11].

Method	LPIPS ↓	FID ↓	FVD ↓
MDGAN [54]	0.49	68.9	385.1
DTVNet [57]	0.35	74.5	693.4
DL [30]	0.41	41.1	351.5
AL [15]	0.26	16.4	307.0
cINN [11]	0.23	10.5	134.4
LaMD (Ours)	0.29	10.3	127.1

We also compare the quantitative results with related work in Tab. 3 and the results show that our model significantly outperforms state-of-the-art methods.

For Landscape, we generate videos with 32 frames at 128×128 resolution following [11, 54, 57]. Our model achieves superior performance when comparing with the state-of-the-art shown in Tab. 4. And the qualitative results are shown in Fig. 7, which validate our model can generate realistic videos. The generated motion not only captures reasonable cloud deformation, but also keeps well consistency during long-range movements.

4.4. Text-Image-to-Video Generation

TI2V is a recently proposed task that involves generating videos from a given image and text descriptions. In contrast to I2V, TI2V requires greater control over the motion, which is specified in detail through text descriptions. To enable this, we concatenate the text embeddings extracted by a text encoder with the content features of the given image as multi-modal conditions in DMG. Hence, the motion latents could retrieve information from both image appearance and text descriptions.

We conduct experiments on CATER-GEN-v1 and CATER-GEN-v2. The generated videos are shown in Fig. 8. With the given image and provided ambiguous descriptions as input, our model successfully generates diverse videos with coherent movements as specified in text. The quantitative results compared to the benchmark MAGE [26] are shown in Tab. 5. Our proposed LaMD significantly improves the generation performance which validates the effectiveness of LaMD to generate highly controllable mo-

Table 5. Quantitative results on CATER-GENs under diverse video generation.

Datasets	Method	LPIPS↓	FID↓	FVD↓
CATER-GEN-v1	MAGE [26]	0.22	62.89	45.49
	LaMD (Ours)	0.08	7.98	19.74
CATER-GEN-v2	MAGE [26]	0.26	39.38	69.44
	LaMD (Ours)	0.13	3.49	13.12

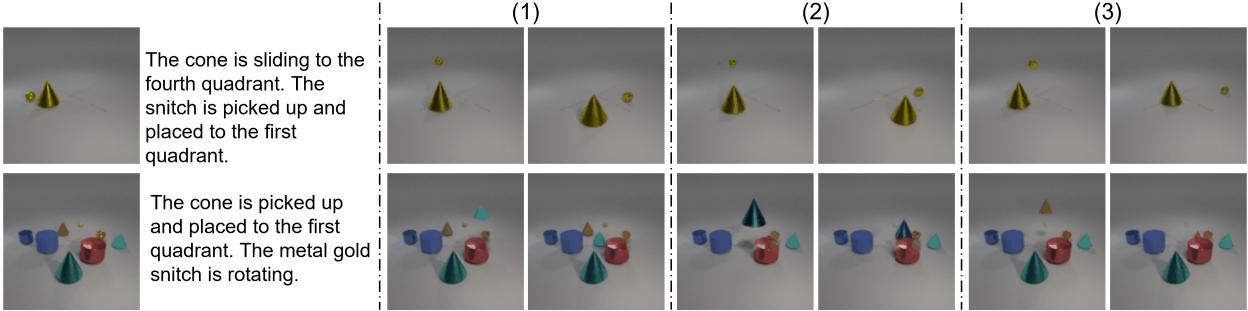


Figure 8. The generated samples on CATER-GEN-v1 and CATER-GEN-v2 datasets under diverse video generation. Given an image and description in the left column, we generate multiple videos and display the 8th and 16th frames of each video in the right three columns.

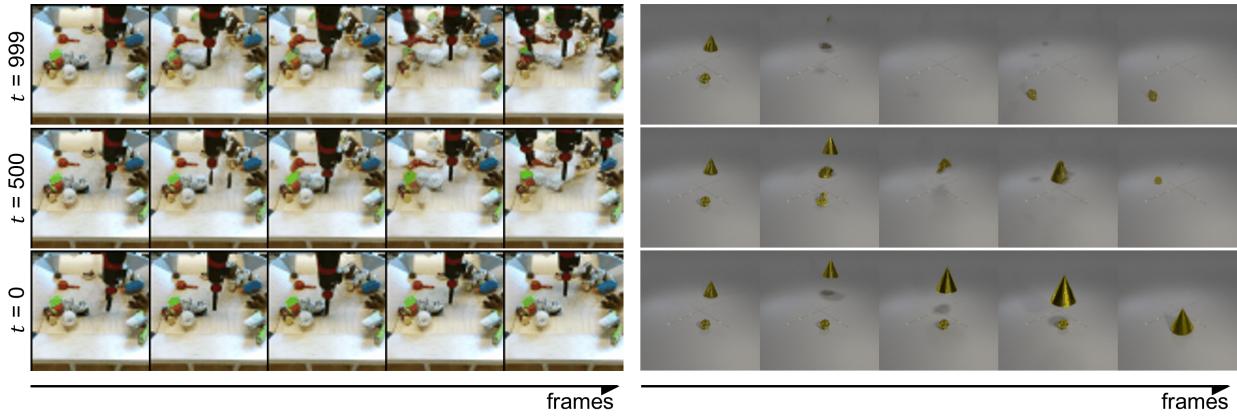


Figure 9. The generated videos at different timesteps during sampling, where $t = 999$ corresponds to synthesized videos based on random latent motion and $t = 0$ represents final generated videos. (The text condition provided in the example shown in the right column is "The cone is picked up and containing the snitch".)

tion.

In addition, we visualize the videos based on generated motion at different timesteps t during sampling process in Fig.9. In the first row, videos are generated based on randomly sampled motion representation from normal Gaussian distribution, which is the initial state of latent motion when $t = 999$. The corresponding synthesized videos only involve chaotic movements of subjects (robot arm in BAIR and objects in CATER-GENs), which suggests that MCD-VAE can effectively extract motion patterns from the data distribution and also validates the decomposition ability of motion. During recursive sampling process, the movements in generated videos are more orderly in pace with the gradual denoising of latent motion, and finally become natural based on final state of latent motion at timestep $t = 0$ in the last row. It validates that generated motion is progressively refined towards the direction of fitting content condition.

5. Conclusion

In summary, we introduced a novel paradigm and corresponding framework LaMD for video generation. By decomposing and compressing motion from videos, the

video generation is refactored into motion generation and video reconstruction. Specifically, MCD-VAE is designed to highly compress motion representation and reconstruct videos through motion-content fusion. The DMG is applied to progressively generate the decomposed motion on continuous latent space. Our framework supports image-driven video generation tasks and achieves much faster sampling speed than other video diffusion models. The experiments demonstrate that LaMD is capable of generating high-quality videos with a wide range of motions, from stochastic dynamics to highly controllable movements. It also achieves new state-of-the-art performance on benchmark datasets on image-to-video and text-image-to-video generation tasks. In the future, we look forward to scale the MCD-VAE to larger video domains and combine it with advanced image generation models, enabling us to tackle boarder applications such as text-to-video generation. Furthermore, there is significant potential in incorporating pre-trained image autoencoder into MCD-VAE to achieve highly compression on representations of both content and motion, which can significantly reduce the resource cost of video generation models.

Appendix

A. Implementation Details

Table 6. Hyperparameters and training details for MCD-VAE. All models trained on 6 NVIDIA GeForce RTX 3090 GPUs.

	BAIR	Landscape	CATER-GEN-v1	CATER-GEN-v2
Resolution	$16 \times 64 \times 64 \times 3$	$32 \times 128 \times 128 \times 3$	$16 \times 128 \times 128 \times 3$	$16 \times 128 \times 128 \times 3$
f_s	4	4	4	4
f_t	16	32	16	16
z_m -shape	$16 \times 16 \times 3$	$32 \times 32 \times 4$	$32 \times 32 \times 1$	$32 \times 32 \times 3$
\mathcal{E}_I -channels	128	128	128	128
\mathcal{E}_M -channels	64	32	32	64
\mathcal{D}_V -channels	128	128	128	128
Channel Multiplier	1,2,4	1,2,4	1,2,4	1,2,4
β	1e-2	1e-5	1e-2	1e-2
Model size	146M	130M	126M	146M
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	5e-5	5e-5	5e-5	5e-5
Training epochs	600	150	170	50

Table 7. Hyperparameters and training details for DMG. All models trained on 6 NVIDIA GeForce RTX 3090 GPUs. The sampling time is measured on a single NVIDIA GeForce RTX 3090 GPU.

	BAIR	Landscape	CATER-GEN-v1	CATER-GEN-v2
z_m -shape	$16 \times 16 \times 3$	$32 \times 32 \times 4$	$32 \times 32 \times 1$	$32 \times 32 \times 3$
Diffusion steps	1000	1000	1000	1000
Noise Schedule	linear	linear	linear	linear
Base channels	192	192	128	128
Channel Multiplier	1,2,4	1,2,4,4	1,2,4	1,2,4
Blocks per resolution	2	2	2	2
Attention Resolution	1,2,4	1,2,4	1,2,4	1,2,4
Conditioning embedding dimension	512	512	512	512
Conditioning transformer dimension	-	-	512	512
Model size	241M	241M	149M	149M
Dropout	-	0.2	0.2	-
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	2e-5	2e-5	2e-5	2e-5
Batchsize	96	96	96	96
Training epochs	400	400	700	400
Sampling timesteps	200	200	200	200
Sampling time (per video)	9s	11s	10s	10s

B. Sample Quality vs. Model Size

Table 8. Comparison of sample quality and sampling speed between different model size on BAIR dataset.

Base Channels	Model Size	Sampling Time (per video)	Sampling time (per step)	FVD ↓
128	109M	9.6s	0.048s	61.3
192	241M	10.1s	0.050s	57.0

C. Additional Qualitative Results

Condition

Generated Videos

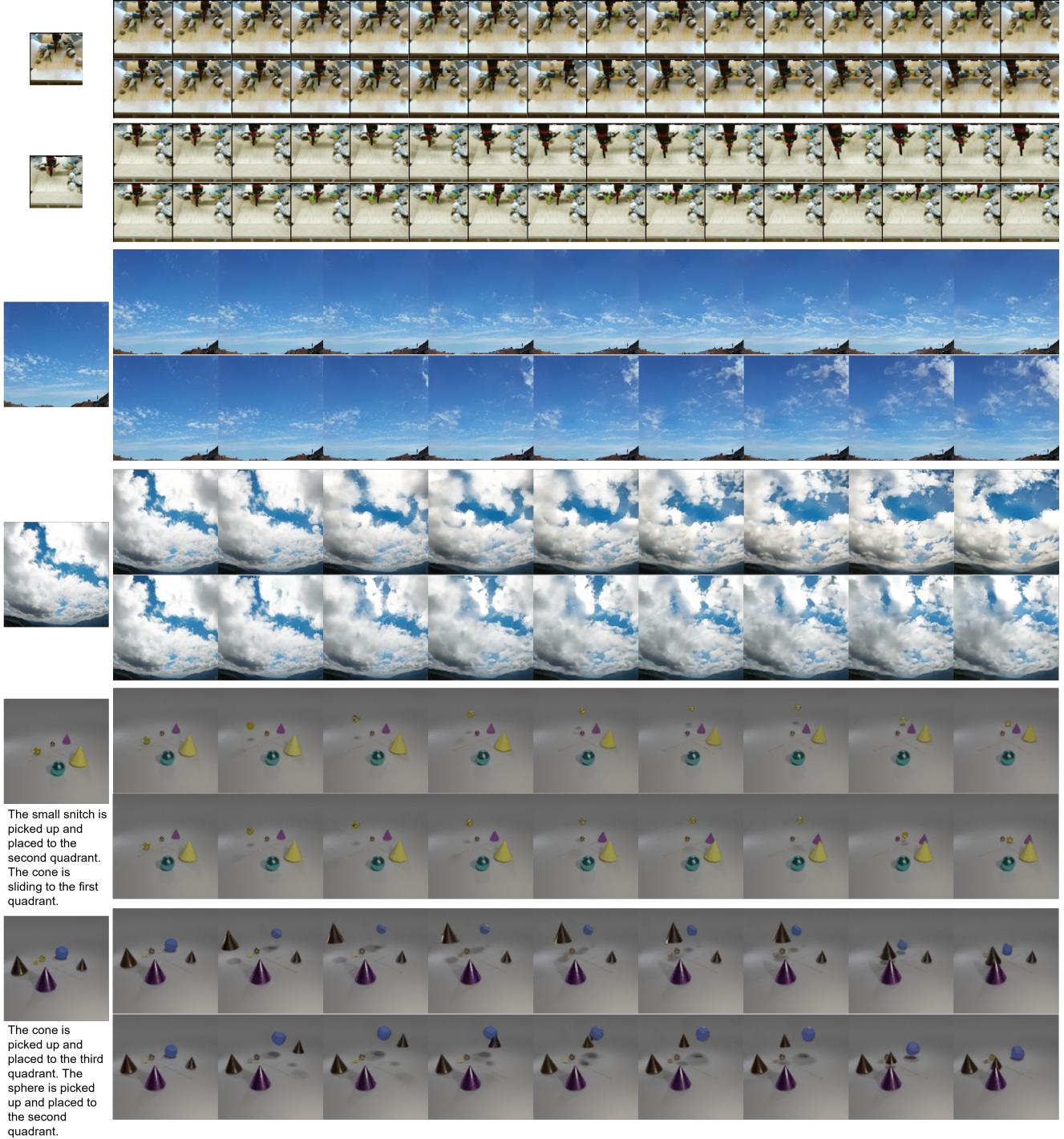


Figure 10. Diverse video generation results (right) under given conditions (left).

References

- [1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017. 3
- [2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2018. 3
- [3] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021. 5, 7
- [4] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Björn Ommer. Understanding object dynamics for interactive image-to-video synthesis. In *CVPR*, pages 5171–5181, 2021. 3
- [5] Lluís Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *ICCV*, pages 7608–7617, 2019. 7
- [6] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, pages 11315–11325, 2022. 3
- [7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In *ICML*, pages 1691–1703. PMLR, 2020. 2
- [8] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *MICCAI*, pages 424–432. Springer, 2016. 3
- [9] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019. 7
- [10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, volume 34, pages 8780–8794, 2021. 2
- [11] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G. Derpanis, and Björn Ommer. Stochastic image-to-video synthesis using cinns. In *CVPR*, pages 3741–3752, 2021. 1, 3, 5, 6, 7
- [12] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *NeurIPS*, page 658–666, 2016. 4
- [13] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *NeurIPS*, 29:658–666, 2016. 6
- [14] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *CoRL*, pages 344–356, 2017. 5
- [15] Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis. *ACM TOG*, 38(6):1–19, 2019. 7
- [16] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 1, 4
- [17] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12868–12878, 2021. 3
- [18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2
- [19] Ligong Han, Jian Ren, Hsin-Ying Lee, Francesco Barbieri, Kyle Olszewski, Shervin Minaee, Dimitris Metaxas, and Sergey Tulyakov. Show me what and tell me how: Video synthesis via multimodal conditioning. In *CVPR*, pages 3615–3625, 2022. 1
- [20] Ligong Han, Jian Ren, Hsin-Ying Lee, Francesco Barbieri, Kyle Olszewski, Shervin Minaee, Dimitris Metaxas, and Sergey Tulyakov. Show me what and tell me how: Video synthesis via multimodal conditioning. In *CVPR*, pages 3615–3625, June 2022. 3
- [21] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022. 1, 3
- [22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 6
- [23] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 3
- [24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pages 6840–6851, 2020. 2, 4
- [25] Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022. 1, 3, 7
- [26] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: controllable image-to-video generation with text descriptions. In *CVPR*, pages 18219–18228, 2022. 5, 7
- [27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2
- [28] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, pages 11523–11532, June 2022. 3
- [29] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *ECCV*, page 609–625, 2018. 1
- [30] Elizaveta Logacheva, Roman Suvorov, Oleg Khomenko, Anton Mashikhin, and Victor Lempitsky. Deeplandscape: Adversarial modeling of landscape videos. In *ECCV*, pages 256–272. Springer, 2020. 7
- [31] Willi Menapace, Stéphane Lathuilière, Sergey Tulyakov, Aliaksandr Siarohin, and Elisa Ricci. Playable video generation. In *CVPR*, pages 10061–10070, 2021. 3

- [32] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171, 2021. 5
- [33] Ruslan Rakhimov, Denis Volkonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. In *VISIGRAPP*, pages 101–112, 2021. 5, 6, 7
- [34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 3
- [35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021. 1
- [36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, volume 139, pages 8821–8831, 18–24 Jul 2021. 3
- [37] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, 2019. 3
- [38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 3, 4, 5
- [39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 3
- [40] Younggyo Seo, Kimin Lee, Fangchen Liu, Stephen James, and Pieter Abbeel. Harp: Autoregressive latent video prediction with high-fidelity image generator. In *ICIP*, pages 3943–3947, 2022. 3
- [41] Shuai Shen, Wenliang Zhao, Zibin Meng, Wanhua Li, Zheng Zhu, Jie Zhou, and Jiwen Lu. DiffTalk: Crafting diffusion models for generalized talking head synthesis. *arXiv preprint arXiv:2301.03786*, 2023. 1
- [42] Gaurav Shrivastava and Abhinav Shrivastava. Diverse video generation using a gaussian process trigger. In *ICLR*, 2021. 3
- [43] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 3
- [44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015. 4
- [45] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, pages 1526–1535, 2018. 3
- [46] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 6
- [47] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NeurIPS*, volume 30, 2017. 3
- [48] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022. 3
- [49] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting video with vqvae. *arXiv preprint arXiv:2103.01950*, 2021. 3
- [50] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991. 3
- [51] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitzia Dantcheva. G3an: Disentangling appearance and motion for video generation. In *CVPR*, pages 5264–5273, 2020. 3
- [52] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*, 2019. 5, 7
- [53] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Dexin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *ECCV*, pages 720–736. Springer, 2022. 3, 7
- [54] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *CVPR*, pages 2364–2373, 2018. 5, 7
- [55] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 1, 3, 6, 7
- [56] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *ICLR*, 2022. 3
- [57] Jiangning Zhang, Chao Xu, Liang Liu, Mengmeng Wang, Xia Wu, Yong Liu, and Yunliang Jiang. Dtvnet: Dynamic time-lapse video generation via single still image. In *ECCV*, pages 300–315. Springer, 2020. 3, 5, 7
- [58] Chuanxia Zheng, Long Tung Vuong, Jianfei Cai, and Dinh Phung. MoVQ: Modulating quantized vectors for high-fidelity image generation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *NeurIPS*, 2022. 3
- [59] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 1