MagicVideo: Efficient Video Generation With Latent Diffusion Models

Daquan Zhou* Weimin Wang* Hanshu Yan
Weiwei Lv Yizhe Zhu Jiashi Feng

ByteDance Inc.
{daquanzhou, weimin.wang, hanshu.yan}@bytedance.com
{vici, yizhe.zhu, jshfeng}@bytedance.com

Abstract

We present an efficient text-to-video generation framework based on latent diffusion models, termed MagicVideo. *Given a text description, MagicVideo can generate photo-realistic video clips with high relevance to the text content. With the proposed efficient latent 3D U-Net design, MagicVideo can generate video clips with 256×256 spatial resolution on a single GPU card, which is 64 × faster than the recent video diffusion model (VDM). Unlike previous works that train video generation from scratch in the RGB space, we propose to generate video clips in a low-dimensional latent space. We further utilize all the convolution operator weights of pre-trained text-to-image generative U-Net models for faster training. To achieve this, we introduce two new designs to adapt the U-Net decoder to video data: a frame-wise lightweight adaptor for the image-to-video distribution adjustment and a directed temporal attention module to capture frame temporal dependencies. The whole generation process is within the low-dimension latent space of a pre-trained variation auto-encoder. We demonstrate that MagicVideo can generate both realistic video content and imaginary content in a photo-realistic style with a trade-off in terms of quality and computational cost. Refer to* https://magicvideo.github.io/# *for more examples.*

1. Introduction

Recent progress on generative models has shown astonishing achievements over a variety of applications such as text-to-image generation [30,37], style transfer [54], image-to-image translation [34], text-to-3D object generation [28] and text-to-video generation [14, 16, 47]. Among them, video generation has shown considerable flexibility and diversity in the generated content over 2D image generation and thus has attracted increasing attention. This work explores text-to-video generation, *i.e.,* generating video clips with contents complying with the provided textual description.

Diffusion models have become increasingly more popular for generative tasks, thanks to their superior generation quality and scaling capability to large datasets. Recent representative text-to-image generative models, such as DALL-E 2 [30], Imagen [37], Parti [52], CogView [16], with large-scale training datasets [39], diffusion-based generative models [12] can generate photo-realistic contents from the given texts.

Despite its recent success in text-to-image generation tasks, the application of diffusion-based generative models for video generation tasks is still under-explored, due to the following difficulties:

- *Data scarcity*. Video data with precise textual descriptions are much harder to collect than image-text data. Unlike images, videos are more difficult to describe with a single text sentence. Besides, each video could contain several clips, with most frames less informative, reducing model learning efficiency.
- *Complex temporal dynamics*. Video data present complex visual dynamics, which are more difficult to learn than still images. Besides the visual contents of every single frame, the temporal consistency among different frames also needs to be modeled, which brings new challenges for generative models.
- *High computation cost*. Each video data may contain hundreds or even thousands of frames. Each frame has a similar computational cost as an image of the same size. Directly processing long videos requires a huge amount of computation and memory cost.

Due to these challenges, recent diffusion-based video generation models propose to deploy a cascaded pipeline [15], which generates low-resolution video frames first, followed

*Equal contribution.

(a) A celebration with Christmas tree and fireworks, starry sky.

(b) Campfire at night in a snowy forest with starry sky in the background.

(c) A 3D model of an elephant origami. Studio lighting.

Figure 1. Qualitative results of our proposed MagicVideo, a text-to-video framework with a latent diffusion model. The figure contains short video clips. Best viewed by Adobe Reader.

by a super-resolution module. Even with this practice, the computational cost is still huge. For example, the computational cost of the denoising decoder alone will take 38G FLOPs for the coarse image generation with 64×64 spatial resolution for a single forward. The forward pass need to be repeated 1000 times for a conventional denoising diffusion model [12], making the whole process extremely expensive on the computation resources. We want to highlight that this process is quadratically proportional to the image resolutions.

The recent latent diffusion model [34] has shown state-of-the-art efficiency on image generation tasks. It first projects the sampled input noise to a latent space with a smaller spatial dimension and then converts to RGB space with larger spatial resolution via a variational auto-encoder [32, 33]. In this way, the spatial dimension of the denoising decoder is kept in a low-dimension space which reduces the computational cost quadratically. For example, with a target image resolution of 256×256, the spatial dimension of the diffusion denoising decoder is reduced to 32×32. Even with the same decoder structure as previous models [30, 37], the computational cost is reduced by $64\times$. This fact motivates us to think of the possibility of generating videos within the latent space.

We adopt latent diffusion models to build our model. Specifically, we develop the video generation algorithm within the latent space of a pre-trained VAE, similar to the recent stable diffusion model for image generation [34]. To address the challenges above for video generation, we introduce the following new model designs that enable us to build the first latent diffusion model for video generation tasks.

To improve data efficiency and relieve the requirement for the video-text paired training data, instead of building our models with 3D convolutions, we choose to adopt 2D convolution together with temporal computation operators to model the spatial and temporal video features. We will detail the model architecture in the following section. With this architecture design, we can re-use the weight parameters of text-to-image models pre-trained on large dataset [39] to initialize the 2D convolutions for the single frame feature processing. Thus we can fully utilize the pre-trained image generation model to generate video frames of moderate quality, even though the video training data is few. A similar practice has been adopted by CogVideo [16], which utilizes the pre-trained weights from CogView [6] for model initialization.

To further reduce the memory cost, we share the same 2D convolutions for processing all the frames. However, this will deteriorate the generation quality of video temporal dynamics (e.g., object motion) because of the feature change across different frames. Therefore, we introduce a new and lightweight *adaptor* module to adjust the distribution of each frame's features. The adaptor uses a small number of learnable scalars to change the distribution since the frames of a video clip are mostly overlapped. Therefore, it is not necessary to use independent neural blocks for frame processing [23]. Our model learns the relation among the frames via a *directed self-attention*: the future frame features are calculated based on all the preceding frames, and the previous frames are unaffected by the future ones. We find this practice improves the motion consistency over conventional self-attention modules as used by existing generative models [30, 34, 37, 40]. Furthermore, the generated video clips can be further smoothed and upsampled via a post-processing frame interpolation model.

We run extensive experiments to show that MagicVideo can generate high-resolution video clips with 64× speed up with a single model, compared to conventional video diffusion models that utilize cascaded frameworks.

We summarize our contributions as follows:

- We propose a new text-to-video generation framework, termed MagicVideo, based on the latent diffusion model. MagicVideo can generate high-resolution video clips (256×256) based on a given text input that is 64× faster than the very recent video diffusion model.
- We propose a new video learning scheme that can process video frames without needing 3D convolution or 2D + 1D convolution block. Instead, we suggest a novel 2D convolution + adaptor block design for video data processing.
- We introduce a simple yet effective directed self-attention module that can learn meaningful motions from the video dataset. We empirically verify that MagicVideo can generate expressive motions with consistent identity across all frames with the proposed directed self-attention.

2. Preliminary

2.1. Generative Modeling through Diffusion Models

Denoising Diffusion Probabilistic Models Deep generative modeling utilizes neural networks to approximate the distribution from which the training data are sampled. Denoising diffusion probabilistic models (DDPM) approximate the probability densities of training data via the reversed processes of Gaussian diffusion processes [12].

For certain data distribution $q(\cdot)$, DDPM approximates the probability density $q(\boldsymbol{x})$ as the marginal of a series of latent variables $x_{0:T}$,

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T} \quad \text{with} \quad \boldsymbol{x} = \boldsymbol{x}_0.$$

This process starts from the standard normal distribution $\mathcal{N}(\cdot; \mathbf{0}, \boldsymbol{I})$ and forms a Markov chain with Gaussian transitions, *i.e.*, $p_\theta(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \boldsymbol{I})$, and

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \equiv \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)).$$

DDPMs use a fixed Markov Gaussian diffusion process, $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$, to approximate the posterior, $p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$, so that we can train DDPMs via likelihood maximization. In specific, we define two seriers, $\alpha_{0:T}$ and $\sigma^2_{0:T}$, where $1 = \alpha_0 > \alpha_1 > \ldots, > \alpha_T \geq 0$ and $0 = \sigma_0^2 < \sigma_1^2 < \cdots < \sigma_T^2$. For any $t > s \geq 0$, we have

$$q(\boldsymbol{x}_t|\boldsymbol{x}_s) = \mathcal{N}(\boldsymbol{x}_t; \alpha_{t|s}\boldsymbol{x}_s, \sigma^2_{t|s}\boldsymbol{I}),$$

where $\alpha_{t|s} = \frac{\alpha_t}{\alpha_s}, \sigma^2_{t|s} = \sigma_t^2 - \alpha^2_{t|s}\sigma_s^2$. Thus,

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t|\alpha_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I}).$$

The parameterized reversed process p_θ of DDPM is optimized by maximizing the associated evidence lower bound (ELBO). By parameterizing the $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ as the form of posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$, the DDPM can be interpreted as iteratively removing noise signals to recover clean signals. The formulation above describes the modeling and training of an unconditional generative model. For the conditional case, the notations share similar forms by simply conditioning on a control signal $\boldsymbol{y}$.

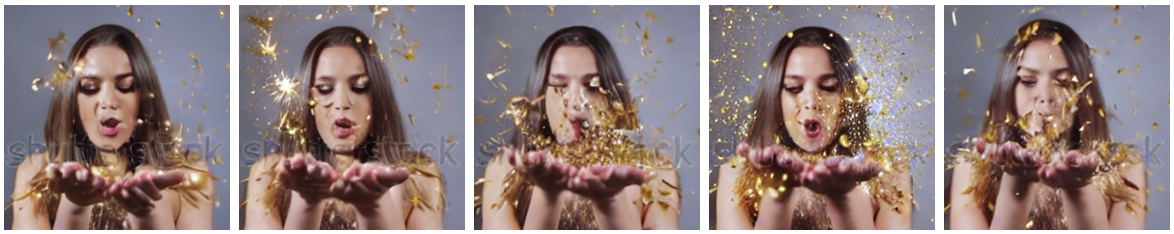
Fast Sampler and Latent Diffusion Models DDPMs have achieved great success in image/video generation [12, 19, 42] and image editing [10, 20, 22, 25, 27]. However, due to the large step number of iterations during sampling, the computational overhead of DDPMs gets very high and impedes their applications. To improve the efficiency of DDPMs, researchers have developed several advanced sampling methods by utilizing high-order SDE/ODE solvers [24, 41, 42]. Besides, researchers also have explored novel diffusion-based modeling methods. Rombach *et al*. 2022 [35] proposed the latent diffusion model (LDM) that models the data distribution in a low-dimensional latent space. Denoising noisy data in a lower dimension may reduce the computational cost in the generation process. Vahdat *et al*. 2021 [46] concurrently proposed a latent score-based model that shares similar ideas with LDM.

Specifically, LDM first trains an autoencoder $\mathcal{E}$-$\mathcal{D}$ to map images $\boldsymbol{x}$ into a low-dimensional space and reconstruct images from latent codes $\boldsymbol{z}$. Then, the autoencoder is jointly trained with a perceptual loss and a patch-based adversarial objective. This training scheme ensures the spatial correspondence between latent codes and the original images. Then, a DDPM with a time-conditional U-Net backbone is used to model the distribution of the latent representations. To enable controllable/conditional generation, LDM uses a domain-specific encoder $\tau(\cdot)$ to project the control signal $\boldsymbol{y}$ (*e.g.* a text prompt) into an intermediate space and subsequently injects the embedding into the U-Net via a cross-attention layer. The implementation of the VAE is inspired by the LDM and based on the public stable-diffusion[1] code base.

2.2. Video Generation

Researchers have explored various video generation methods in the past several years, including using GAN-based [5, 45, 49] and auto-regressive methods to model the

[1]Stable-Diffusion: https://huggingface.co/CompVis/stable-diffusion



Young attractive woman blowing golden confetti from hands. slow motion.

A 3D model of an elephant origami. Studio lighting.

Melting pistachio ice cream dripping down the cone.

A tiger with fur made out of electricity, digital art.

A bunch of colorful candies falling into a tray in the shape of text 'Home'. Smooth video.

A celebration with Christmas tree and fireworks.

Figure 2. Videos generated from various text prompts. MagicVideo produces diverse and temporally-coherent videos that are well-aligned with the given prompt. Note that the videos are generated with base model directly without super-resolution model.

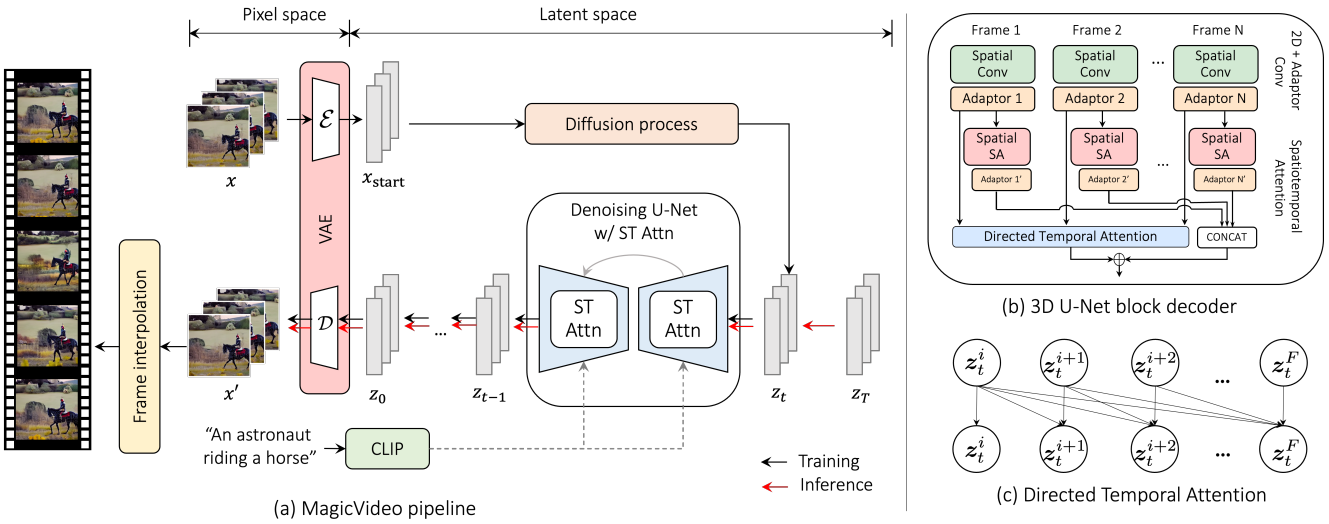


Figure 3. **The overall framework of MagicVideo.** (a) The data flow of both the training and inference phases: during the training phase, a timestep t will be sampled randomly from $[0, T]$ and the input video frames are corrupted via the diffusion process, and a U-Net decoder is used to learn to reconstruct the video frames. During inference phase, a gaussian noise is randomly sampled, and the denoising process is repeated for T times. The denoised latent vector $\boldsymbol{z}$ is then fed into a VAE decoder and converted to the RGB space. (b) The structure of the spatiotemporal attention (ST-Attn) module. (c) The directed attention used in the ST-Attn.

distributions [2, 17, 18, 21, 26, 31] of video frames in RGB or latent spaces. Recently, diffusion-based generative models have shown excellent performance for (conditional) image generation, which triggers significant interest in exploring their applications in video modeling.

Ho *et al*. [14] present a natural extension of the vanilla DDPM for image generation to the video domain. They proposed a 3D U-Net diffusion model architecture and a novel conditional sampling technique for longer video generation. Harvey *et al*. [9] propose to model the conditional distribution of subsequent video frames given certain observed ones so that a long video can be synthesized by conditional sampling in an auto-regressive manner. Similarly, Yang *et al*. [51] propose a two-step framework for video generation—using a deterministic model to predict the next frame given observed ones, then utilizing a stochastic diffusion model to synthesize the residual for correcting the next frame prediction.

In this work, we are interested in text-conditional video generation. Ho *et al*. [11] proposed a cascaded pipeline, termed Imagen Video, to synthesize high-definition videos. The pipeline consists of one base text-to-video module, three spatial super-resolution (SSR), and three temporal super-resolution modules (TSR). The base text-to-video module utilizes temporal attention layers, while temporal convolutions are used in SSR and TSR modules. Concurrently, Singer *et al*. [40] propose a multi-stage text-to-video generation method, termed Make-A-Video. The proposed method exploits a text-to-image model to generate image embeddings, then trains a low-resolution video generation model with conditioning on the image embeddings, and finally trains spatial and spatial-temporal super-resolution models to synthesize high-definition videos. Both Imagen Video and Make-A-Video model the video distribution in the RGB space. Differently, our work explores a more efficient way for video generation by synthesizing videos in a low-dimensional latent space.

3. Methods

In this section, we illustrate the proposed MagicVideo framework in detail. The framework includes three steps: keyframe generation, frame interpolation, and super-resolution. For the keyframe generation, we first present how we modify the 2D convolution blocks with weights pre-trained on the text-image dataset to adapt to the 3D video dataset via a new *adaptor* module (Sec. 3.2.1). Then, we show a novel directed self-attention module that enables the model to learn the motions among frames within a video clip (Sec. 3.2.2). In Sec. 3.3, we show how we interpolate the frames to make the generated smoothing. Finally, we explain how to increase the spatial resolution via a separately trained super-resolution model as detailed in Sec. 3.4. Fig. 3 illustrates the whole pipeline.

3.1. Notions

This paper uses $\boldsymbol{x}_t$ to denote a sequence of video frames corrupted with Gaussian noise at intermediate time step t. $\boldsymbol{x}_t$ is short for $\boldsymbol{x}_t = [\boldsymbol{x}_t^1, ..., \boldsymbol{x}_t^F]$, where $\boldsymbol{x}_t^i$ represents the i^{th}

frame in the sequence. The encoder and decoder of the variational auto-encoder are denoted by $\mathcal{E}(\cdot)$ and $\mathcal{D}(\cdot)$, respectively. The video frames are mapped into the latent space one by one, *i.e.*, $\boldsymbol{z}_t = [\mathcal{E}(\boldsymbol{x}_t^1), ..., \mathcal{E}(\boldsymbol{x}_t^F)]$. We use CLIP [29] to encode the given text prompt $\boldsymbol{y}$, and the obtained embedding is denoted as $\tau(\boldsymbol{y})$. We use $\epsilon_\theta(\boldsymbol{z}_t, t, \tau(\boldsymbol{y}))$ to denote the denoiser of the diffusion model in the latent space.

3.2. Key Frame Generation

Following previous works [15, 16, 40], we generate 16 key-frames and then use a separate model to interpolate these frames to form a long video sequence. During training, we first project the input images into a latent space via a pre-trained variational auto-encoder (VAE) model[2]. We then use a Gaussian diffusion process to corrupt the input frames with a randomly sampled time step $t \sim [1, T]$ with T being the number of total diffusion steps. We then design a novel 3D U-Net decoder, ϵ_θ, to denoise the corrupted frames $\boldsymbol{x}_t^{[F]}$ as detailed below.

3.2.1 2D Convolution with distribution adaptor

The conventional operator unused in the denoising decoder for video data processing is the 3D convolution [4]. However, the computation complexity and hardware compatibility of 3D convolution are significantly worse than that of 2D convolution. Thus, to reduce the high computational cost and redundancy, recent video processing models typically replace 3D convolution with a 2D convolution along the spatial dimension followed by a 1D convolution [44] along the temporal dimension (termed "2D+1D").

In this work, we further simplify this process from the form of "2D+1D" to "2D+adaptor", where the *adaptor* is an even simpler operator compared to the 1D convolution. Specifically, given a set of F video frames, we apply a shared 2D convolution for all the frames to extract their spatial features. After that, we assign a set of distribution adjustment parameters to adjust the mean and variance for the intermediate features of every single frame via:

$$\boldsymbol{z}_t^i = S^i \cdot \text{Conv2d}(\boldsymbol{z}_t^i) + B^i, \tag{1}$$

where x_t^i denotes the feature of the i^{th} frame at denoising time step t, and $S, B \in \mathbb{R}^{F \times C}$ are two groups of learnable parameters used for variance adjustment and shift adjustment of the extracted features $\boldsymbol{z}_t$. This design is based on the observation that the frames within each video clip are semantically similar. The small difference among frames may not be necessary for a dedicated 1D convolution layer. Instead, we model those differences via a small group of parameters. The details of the adaptor are shown in Fig. 3(b).

[2]The weights are taken from https://github.com/CompVis/stable-diffusion

3.2.2 Spatial and directed temporal attention

We introduce a new directed self-attention to better model the video temporal dynamics for the denoising decoder. Following previous works [14, 30, 37], we adopt self-attention modules after the down-sampling blocks with $4\times$, $8\times$ and $16\times$ spatial reduction with the convolution blocks. The attention module is conducted along the spatial and temporal dimensions separately. The output of the two parallel attention modules is added and passed to the following modules:

$$\boldsymbol{z}_t = \text{S-Attn}(\boldsymbol{z}_{t-1}) + \text{T-Attn}(\boldsymbol{z}_{t-1}), \tag{2}$$

where S-Attn denotes the attention calculated along the spatial dimension (i.e., to aggregate the frame-wise feature tokens), and T-Attn denotes the self-attention conducted along the temporal dimension. More concretely, the spatial attention is calculated following previous works [30, 37] via:

$$\text{S-Attn} = \text{Cross-Attn}(\text{LN}(\text{MHSA}(\text{LN}(\boldsymbol{z}_{t-1})))), \tag{3}$$

where MHSA is a standard Multi-head Self-attention module used in vision transformers [7, 53]. LN denotes the layer normalization [1]. Finally, cross-Attn denotes the cross self-attention module where the attention matrix is calculated between the frame tokens $\boldsymbol{z}_{t-1}$ and the text embedding $\boldsymbol{y}$. However, different from recent VDM [14], we do not apply the self-attention along the temporal dimension directed. Instead, we propose a new directed self-attention as detailed below.

Directed temporal attention. Many recent video generation frameworks [14, 40] use a conventional (i.e., bi-directional) self-attention along the temporal dimension for the motion learning in the video dataset. We notice that the self-attention matrix missed a critical feature of the video data: the motions are directional. In video data, the frames are expected to change in a regular pattern along the temporal dimension. We propose a directed self-attention mechanism to inject the temporal dependency among the frames.

Specifically, with a set of given video frames features, $\boldsymbol{z}_t \in \mathbb{R}^{F \times C \times H \times W}$ where C, F, H, W denotes the batch size, number of feature channels, number of frames and the spatial dimension of the features respectively. We first reshape $\boldsymbol{z}_t$ into shape $HW \times \textit{\#Heads} \times F \times \frac{C}{\textit{\#Heads}}$ and treat each pixel of each frame as a token, where *#Heads* denotes the number of attention heads. Temporal attention is applied to the tokens of the exact spatial location across different frames to model their dynamics. Specifically, we obtain their query Q_t, key K_t, and value V_t embeddings used in the self-attention via three linear transformations. We calculate the temporal attention matrix, $A_t \in \mathbb{R}^{\textit{\#Heads} \times F \times F}$ via:

$$A_t = \text{Softmax}(Q_t K_t^\top / \sqrt{d}) \odot M, \tag{4}$$

where d is the dimension of embeddings per head and M is an lower triangular matrix with $M_{p,q} = 0$ if $p > q$ else 1. With the implementation of the mask, the present token is only affected by the previous tokens and independent from the future tokens since the frames are arranged based on their temporal sequence during the temporal attention calculation. Fig. 3(c) illustrates this process.

3.2.3 Video Frame Sampling

Unlike the image dataset, each video clip contains a wide range of total frame numbers, from hundreds to thousands of frames. Due to the considerable computational and memory cost, it is impractical to process all frames within each video clip with a single forward pass. Thus, a typical practice is to sample a small subset and use it to represent the whole video clip [14, 16]. However, a unified sampling strategy would result in the same subset of sample keyframes containing different amounts of information and thus increase the difficulty of training. To ease the training and make the generation more controllable, we use a sampling strategy to randomly sample a small portion of the video clip and then uniformly sample 16 frames within the selected subset of the video clip for training. We then calculate a corresponding frame-per-second (FPS), denoted as ν, based on the sampling frames and add it to the data features. Specifically, during training, we will first fetch the FPS of the video data from the metadata and then use two linear layers to transform it into an embedding:

$$\text{emb}_\nu = \text{Linear}(\text{SiLU}(\text{Linear}(\text{Sin}(\nu)))), \tag{5}$$

where $\text{Sin}(\cdot)$ denotes the sinusoidal position embedding as introduced in [48]. It converts the FPS into an embedding of dimension C. SiLU denotes the sigmoid ReLU [8]. The embedding $\text{emb}_\nu \in \mathbb{R}^C$ will be added to the video frame features, $\boldsymbol{z} \in \mathbb{R}^{F\times H\times W\times C}$. During training, one can control the frame smoothness by choosing different FPS values.

3.2.4 Training Objective

We directly use frame reconstruction loss for the model training. Specifically, given a sequence of video frames, the loss per sequence is computed as follows,

$$\mathcal{L}(\boldsymbol{x}_0) = \mathbb{E}_t \sum_{i=0}^{F} \|\boldsymbol{x}_0^i - \epsilon_\theta(\boldsymbol{z}_t^i, t, \boldsymbol{y})\|_2^2. \tag{6}$$

We empirically verify that this simple reconstruction is enough to train the model.

3.3. Frame interpolation

To increase the temporal resolution and smooth the generated video, we use a separate frame interpolation network to insert new frames between adjacent keyframes. We also train the interpolation model in the latent space in a similar pipeline as the keyframe generation framework. The main difference is that the latent variable $\boldsymbol{z}$ is conditioned on the adjacent two frames. Note that the condition frames are also projected into the latent space via the same VAE used for the keyframe generation. We further found it helpful to spherically interpolate a third latent vector from the two conditional latent vectors and concatenate it into the input, which changes the input channels to 16.

3.4. Super-resolution

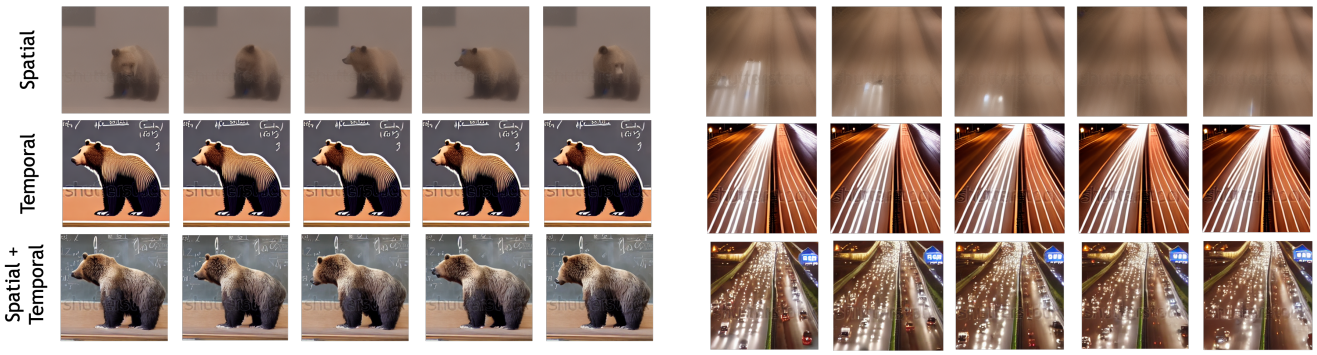
To generate high-resolution videos, we trained a diffusion-based super-resolution model [38] (SR) on pixel space to upsample 256x256 to 1024x1024 resolution. The SR model is trained only on image datasets because large-scale high-resolution video datasets are not publicly available and are extremely hard to collect. In contrast, large-scale high-resolution image datasets are much easier to obtain. In our experiments, we observed the SR model trained on images could also perform well on the videos.

To reduce its computation and VRAM, we train the model on 512×512 random crops of 1024×1024 images for 1M iterations. Chitwan et al. [36] observed noise conditioning augmentation on super-resolution is critical for generating high-fidelity images. Therefore, following the practice of [36], we added the Gaussian noise of a random level to the low-resolution image and condition the diffusion model on the noise level.

4. Experiments

Datasets. We take the pre-trained weights based on LDM [34] on Laion 5B [39] and use it as the initialization of the video 3D U-net denoising decoder. We then fine-tune the video generation model on Webvid-10M for 100k iterations. When comparing with other methods quantitatively, we test the zero-shot generation performance with text prompt from UCF-101 [43], and MSR-VTT [50] and calculate the FID and FVD based on their test data.

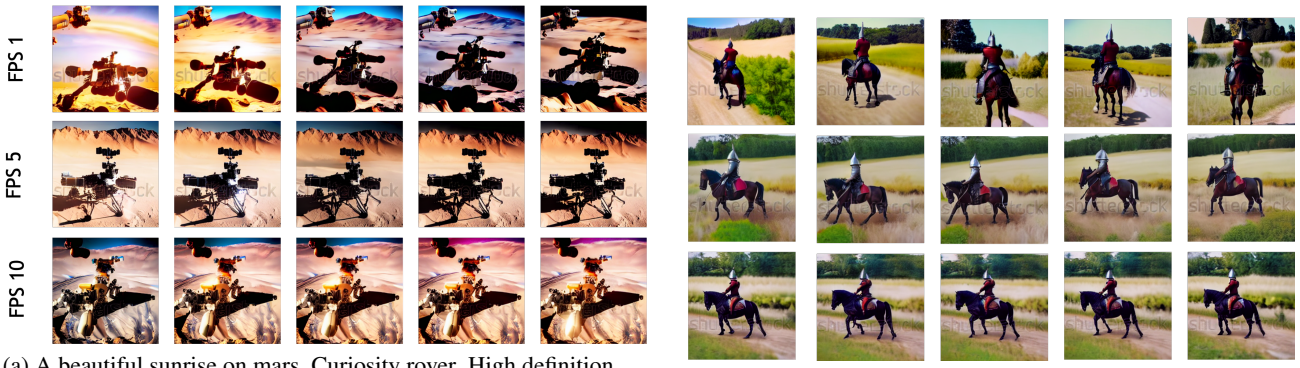
Implementation details. Following VDM [15], we use noisy conditioning augmentation during training. Specifically, following [13], we randomly sample a signal-to-noise ratio (SNR) and Gaussian Noise to adjust the text condition embeddings and the loss scale. For frame interpolation model training, we first uniformly sample a subset counting 5% to 15% of the full video clip from the training data. After that, we uniformly select three frames and use them for training. Next, the first and third frames are concatenated to the second frame along the channel dimension. Finally, the reconstruction loss is computed based on the second frame. During the sampling phase, we first generate 16 key-frames and then interpolate them twice with a total number of 61



(a) A confused grizzly bear in calculus class.

(b) Busy freeway at night.

Figure 4. **Illustration of the attention block design.**



(a) A beautiful sunrise on mars, Curiosity rover. High definition, timelapse, dramatic colors.

(b) A knight riding on a horse through the countryside.

Figure 5. **Video smoothness control via FPS embedding.**

frames with a spatial resolution of 256 $\times$ 256 for each input text. The spatial resolution is then further increased to 1024 $\times$ 1024 via a separate super-resolution network.

4.1. Results

Qualitative performance. We first evaluate our video generation model on qualitative generation performance and compare it with recent state-of-the-art generative models. Then, as shown in Fig. 6, we compare the generated video frames with three current strong baselines. We want to highlight that Make-A-Video is a concurrent work just released recently. Compared with CogVideo [16] and VDM [15], both Make-A-Video and our model can generate videos with richer details. For example, with 'Busy freeway at night as the text input, CogVideo and VDM's results only shows abstract scenes with motion flow without any clear objects (e.g., the cars). Differently, Make-A-Video and our MagicVideo can generate complex highway objects such as cars with headlights. However, our framework can even consider distance information with a more photo-realistic camera view. For example, our model shows clearer vehicles near the camera view.

Zero-shot generation performance. We also evaluate MagicVideo quantitatively. Specifically, we use the model pre-trained on the Webvid-10M dataset and use the text descriptions of the test data of the MSR-VTT dataset and the class label of the UCF-101 validation dataset as the conditions and generate 16 key-frames for each text prompt without fine-tuning. The comparison between MagicVideo and other recent SOTA methods is shown in Tab. 1.

4.2. Analysis

Spatiotemporal attention. Different from concurrent works [40], we use a dual self-attention design in parallel: one for spatial self-attention and temporal attention learning in parallel. The detailed architecture for the attention block is shown in Fig. 3(c). To understand the difference between spatial and temporal attention, we visualize the extracted features from each branch separately, and the results are shown in Fig. 4. We found that spatial attention learns diverse video frames, and temporal attention generates consistent output among the frames. Combining the spatial and temporal attention thus considers diversity and consistency for the generated frames.

Figure 6. Qualitative results comparison with recent/concurrent methods. We compare the generation performance of MagicVideo with three recent strong methods: VDM [14], CogVideo [16], and Make-A-Video [40]. The sample videos for all methods are generated with text input 'Busy freeway at night'. The samples of the VDM are taken from Make-A-Video [40].

Table 1. Video generation evaluation on MSR-VTT and UCF-101.

Method	Zero-Shot	#Samples	MSR-VTT (FVD ↓)	UCF-101 (FVD ↓)
CogVideo [16] (Chinese)	Yes	16	–	751
CogVideo [16] (English)	Yes	16	1294	702
MagicVideo (ours)	Yes	16	1290	699

Video smoothness control. As mentioned in Sec. 3.2.3, we add the embedding of fps to the tokens to control the smoothness of the generated videos. To verify its effectiveness, we generate a group of 16 video key frames with different fps values. The results are shown in Fig. 5. When a low FPS value is specified, the distance between adjacent frames is more significant. Thus the overall action space contained by the generated video clips is expected to be more comprehensive. In contrast, when a smaller FPS is specified, the action space contained by the generated video clips is narrower but looks more smooth.

5. Conclusions

Video generation is an open research challenge. In this work, we made a step toward solving this challenge. In particular, we focused on improving the data and computational efficiency of the video generation models. We leveraged the recent latent diffusion model and developed the video generation framework, MagicVideo, in a low-dimensional latent space. Additionally, we introduced several new designs, including the directional attention, and the adaptor module, to sufficiently utilize pre-trained image generation models. Finally, we demonstrated MagicVideo indeed could generate photo-realistic videos from a text description efficiently.

Ethical impact. Video generation can have significant ethical impacts. Besides the applications on AI assistant content generation (AIGC) for entertainment and art creation, video generation methods are also applicable for malicious purposes by editing videos. However, the fake contents can be detected with current deep fake detection technology. Another potential issue is that we use the pre-trained weights from Stable Diffusion v1.3 [34], which was trained on the LAION dataset [39]. Therefore, it may inherit the LAION dataset contents that have ethical issues [3].

Acknowledgement. We want to thank Dr. Jun Hao Liew for his suggestions on the figures of this work.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. FitVid: Overfitting in Pixel-Level Video Prediction, June 2021. arXiv:2106.13195 [cs].

[3] Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*, 2021.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[5] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial Video Generation on Complex Datasets, Sept. 2019. arXiv:1907.06571 [cs, stat].

[6] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *arXiv preprint arXiv:2204.14217*, 2022.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[9] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible Diffusion Modeling of Long Videos. Technical Report arXiv:2205.11495, arXiv, May 2022. arXiv:2205.11495 [cs] type: article.

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control, Aug. 2022. arXiv:2208.01626 [cs].

[11] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen Video: High Definition Video Generation with Diffusion Models, Oct. 2022. arXiv:2210.02303 [cs].

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, Dec. 2020. arXiv:2006.11239 [cs, stat].

[13] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47–1, 2022.

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. Technical Report arXiv:2204.03458, arXiv, June 2022. arXiv:2204.03458 [cs] type: article.

[15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[16] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers, 2022.

[17] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. Technical Report arXiv:2205.15868, arXiv, May 2022. arXiv:2205.15868 [cs] type: article.

[18] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video Pixel Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1771–1779. PMLR, July 2017. ISSN: 2640-3498.

[19] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising Diffusion Restoration Models, Feb. 2022. arXiv:2201.11793 [cs, eess].

[20] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-Based Real Image Editing with Diffusion Models, Oct. 2022. arXiv:2210.09276 [cs].

[21] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. VideoFlow: A Conditional Flow-Based Model for Stochastic Video Generation. Mar. 2020.

[22] Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. MagicMix: Semantic Mixing with Diffusion Models, Oct. 2022. arXiv:2210.16056 [cs].

[23] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. *arXiv preprint arXiv:2208.03550*, 2022.

[24] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps, Aug. 2022. arXiv:2206.00927 [cs, stat].

[25] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting Using Denoising Diffusion Probabilistic Models. page 11.

[26] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error, Feb. 2016. arXiv:1511.05440 [cs, stat].

[27] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. Technical Report arXiv:2108.01073, arXiv, Jan. 2022. arXiv:2108.01073 [cs] type: article.

[28] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021.

[30] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[31] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos, May 2016. arXiv:1412.6604 [cs].

[32] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[33] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[34] 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.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models, Apr. 2022. arXiv:2112.10752 [cs].

[36] 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. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[37] 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, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

[38] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[39] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Theo Coombes, Cade Gordon, Aarush Katta, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: laion-5b: A new era of open large-scale multi-modal datasets. https://laion.ai/laion-5b-a-new-era-of-open-large-scale-multi-modal-datasets/, 2022.

[40] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-A-Video: Text-to-Video Generation without Text-Video Data, Sept. 2022. arXiv:2209.14792 [cs].

[41] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models, June 2022. arXiv:2010.02502 [cs].

[42] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS. page 36, 2021.

[43] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[44] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[45] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing Motion and Content for Video Generation, Dec. 2017. arXiv:1707.04993 [cs].

[46] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based Generative Modeling in Latent Space. Technical Report arXiv:2106.05931, arXiv, Dec. 2021. arXiv:2106.05931 [cs, stat] version: 3 type: article.

[47] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[49] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating Videos with Scene Dynamics, Oct. 2016. arXiv:1609.02612 [cs].

[50] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016.

[51] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion Probabilistic Modeling for Video Generation. Technical Report arXiv:2203.09481, arXiv, May 2022. arXiv:2203.09481 [cs, stat] type: article.

[52] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation, 2022.

[53] Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M Alvarez. Understanding the robustness in vision transformers. In *International Conference on Machine Learning*, pages 27378–27394. PMLR, 2022.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Aug. 2020. arXiv:1703.10593 [cs].