Simple and Temporal Latent Diffusion Framework for Text to Video Generation

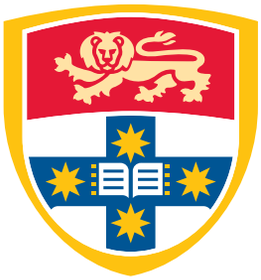
ENZE REN
SID: 520638064

Supervisor: Prof. Zhiyong Wang

This thesis is submitted in partial fulfillment of
the requirements for the Research Pathway Project

School of Computer Science
The University of Sydney
Australia

24 May 2024



THE UNIVERSITY OF
SYDNEY

Student Plagiarism: Compliance Statement

I certify that:

I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to the University commencing proceedings against me for potential student misconduct under Chapter 8 of the University of Sydney By-Law 1999 (as amended);

This Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

Name: Enze Ren

Signature:  **Date**: 24 May 2024

Abstract

Text-to-Video (T2V) generation is currently a challenging task. Due to the successful impact of Text-to-Image (T2I) generation, many recent works attempt to transfer this success to T2V generation. However, previous methods not only require a large number of video-text datasets for training and the T2V generation model contains huge parameters, but also find it difficult to achieve temporal consistency. The Latent Diffusion Model (LDM) can generate high-resolution images in latent space. Inspired by it, we propose a method that extends LDM to a T2V generation model through only pre-trained LDM. The pre-trained T2I model can produce excellent static images, but cannot generate time series. We also propose a customized temporal attention and an effective temporal convolution technique. These two modules enable the LDM to learn the temporal relationship between video frames, in order to obtain more information about temporal consistency, which enable to generate temporal consistent videos with less computational resources. The results indicate that our method has the least number of parameters. A large number of qualitative and evaluative experiments have shown that our method has good performance.

Acknowledgements

First, I would like to appreciate Prof. Zhiyong Wang for giving me the opportunity to work on this research, and assistance, supervision throughout the project.

I would like to thank Zhuqiang Lu and Kun Hu for their help in this project. They provide me with relevant expertise and make a major influence on the direction of the project. In particular, the weekly meetings they organized and the privately organized meetings were extremely helpful to me.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

In recent years, generative artificial intelligence becomes a popular computer vision research area. The research from Text-to-Image (T2I) generation to Text-to-Video (T2V) generation go through many model updates and alternations. The T2I generation models in the earlier years are mainly based on Generative Adversarial Networks (GAN) (Goodfellow et al., 2014; Karras et al., 2019, 2020, 2021; Sauer et al., 2022) and Autoregressive Transformers (Esser et al., 2021; Ramesh et al., 2021; Yu et al., 2022a), which are followed by T2V generation methods (Babaeizadeh et al., 2018; Brooks et al., 2022; Castrejon et al., 2019; Denton and Fergus, 2018; Franceschi et al., 2020; Ge et al., 2022; Gupta et al., 2018, 2022; Hong et al., 2022; Kahembwe and Ramamoorthy, 2020; Lee et al., 2018; Li et al., 2018; Luc et al., 2020; Marwah et al., 2017; Mittal et al., 2017; Pan et al., 2017; Saito et al., 2020; Skorokhodov et al., 2022; Tian et al., 2021; Villegas et al., 2017; Vondrick et al., 2016; Weissenborn et al., 2019; Wu et al., 2021, 2022; Yan et al., 2021; Yu et al., 2022b) with the foundation of them. Then with the advent of diffusion models, it become the primary implementation of generative AI. Many of the recent top performing T2I generation models (Dhariwal and Nichol, 2021; Feng et al., 2023; Ho et al., 2020, 2022b; Liu et al., 2022; Nichol and Dhariwal, 2021; Nichol et al., 2022; Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022; Song et al., 2020b; Xue et al., 2024) and T2V generation models (Harvey et al., 2022; Ho et al., 2022c; Luo et al., 2023; Höppe et al., 2022; Voleti et al., 2022; Yang et al., 2022) are implemented based on diffusion models. This is due to the fact that diffusion models can generate relatively high quality images with high diversity. However, despite the great success of diffusion models in T2I generation, they face many problems in the field of T2V generation. Most of the T2V generation methods are trained based on large-scale public text-video datasets. Such methods consume huge computational resources due to the presence of large-scale video-text datasets and huge diffusion model parameters. On the other hand, the generated videos usually lack temporal consistency. This is because the extra temporal dimension of videos compared to images. Nowadays, T2V generation methods generally suffer from high computational consumption and poor temporal consistency. Here,

we retain the idea of using pre-trained T2I models. This is because pre-trained T2I models already learn very good spatial modeling. T2V models are constructed by simply learning additional temporal modeling. This approach focuses on how to achieve temporal consistency T2V generation using less computational resources.

We implement our T2V model based on the Latent Diffusion Model (LDM) (Rombach et al., 2022). The benefit of doing so has two aspects. First, the LDM is already capable of generating high-resolution images. This is because the image-text dataset is richer than the video-text dataset. Therefore, LDM trained on image-text dataset performs better. The second aspect is that using LDM can not learn spatial content. The model can put all attention on learning the video dynamics. We only need to add extra temporal modules on the pre-trained LDM to achieve good T2V generation. The extra temporal module can control the motion generation reasonably well. The temporal module supplements the information of temporal consistency on top of the still images generated by the LDM. That is, the temporal module complements the LDM spatial modeling. This enables our model to not need to learn video generation from scratch, which greatly reduces the consumption of resources. Therefore, we propose a diffusion-based T2V generation method. It is based on a pre-trained LDM. All parameters are frozen except for the added temporal modules, preserving the ability of the original pre-trained model to generate images. Since the LDM generates images in batches and the batches are disjoint, the temporal modules enable the samples from each batch to be temporally aligned. Our temporal modules contain a temporal convolution and a temporal attention. The temporal convolution is a 1D convolution that learns temporal information with very few parameters. The temporal attention is called Temporal First-Adjacent Attention (TFAA). Compared to other methods (An et al., 2023; Blattmann et al., 2023; Ge et al., 2023; Ho et al., 2022a; Luo et al., 2023; Wang et al., 2023; Zhou et al., 2022), TFAA consumes less resources to achieve the same performance. We train the temporal modules on video-text dataset to get temporally consistent LDM model. A simple temporal consistency T2V generation model is finally obtained.

Our contributions are summarized as follows:

- We propose a T2V framework based on pre-trained LDM. The framework freezes all parameters of T2I for training.
- We propose a 1D temporal convolution layer and a uniquely structured temporal attention layer TFAA. These two temporal modules have less number of parameters to achieve temporal consistency compared to other methods.

- We evaluate and perform ablation studies in comparison with other methods. The results show that our method uses less parameters to achieve good performance.

CHAPTER 2

Literature Review

2.1 Text-to-Image Generation

Generative Adversarial Network. Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is the widely used method to achieve Text-to-Image (T2I) generation in early years. Generative Adversarial What-Where Network (GAWWN) (Reed et al., 2016) proposes a conditional coding method, which uses the precise prompt input to generate the images. After that, Stacked Generative Adversarial Networks (StackGAN) (Zhang et al., 2017) and Stacked Generative Adversarial Networks++ (StackGAN++) (Zhang et al., 2018) are proposed. By dividing multiple issues into several smaller problems, StackGAN and StackGAN++ propose a different method to stack with the optimisation generator, where they serialise GANs in a stack manner. In order to achieve a higher resolution of images, the follow-up GAN-based methods are trained on the pre-order model, which achieve the high-resolution T2I generation in a distributed manner.

In order to solve the issue of image detail loss, Attentional Generative Adversarial Network (AttnGAN) (Xu et al., 2018) further interprets prompt description and combines conditional coding with image production in an attention-grabbing manner. Another challenging issue of GAN is mode collapse, which means that the model consistently produces repeated images with low diversity. In order to solve this problem, Wasserstein Generative Adversarial Networks (WGAN) (Arjovsky et al., 2017) proposes a new loss function to achieve T2I generation. Due to the propose of GAN-based Pixel2Pixel (Isola et al., 2017) and Cycle-Consistent Generative Adversarial Networks (CycleGAN) (Zhu et al., 2017), which have realised a range of image-generating applications such texture modification and style transfer, T2I generation has become much more practical. Progressive Generative Adversarial Networks (ProgressiveGAN) is proposed by NVIDIA (Karras et al., 2018), which generates previously high-definition images by starting with low-resolution images and progressively growing the neural network's size. Style-Based Generative Adversarial Networks (StyleGAN) (Karras et al., 2019, 2020, 2021) is made

possible thanks in part to ProgressiveGAN. The fundamental component of the StyleGAN series is Style Modulation, which is introduced via the normalisation layer into the generation process to generate high-quality images and controlled hierarchical features.

Diffusion Models. Recently, generative semantic and combinatorial capabilities of Diffusion Models (DMs) (Sohl-Dickstein et al., 2015) have been shown, drawing interest from academia area. The University of California, Berkeley in the United States proposes the Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020). DDPM enhances the model representation and significantly lowers the training difficulty. Nonetheless, the primary barrier impeding the advancement of DDPM is its inadequate generating efficiency. Improved DDPM (Nichol and Dhariwal, 2021) and Denoising Diffusion Implicit Models (DDIM) (Song et al., 2020a) are both optimized and accelerated. To produce high-quality images, DDIM balances sampling efficiency and variety. Leveraging the powerful features of CLIP (Radford et al., 2021), DMs are effectively used in T2I generation. With its astounding image creation outputs, the Disco diffusion model usher in the AIGC era of diffusion modelling. However, it has issues with lengthy generating times and stunning images with insufficient information. The DALL-E2 (Ramesh et al., 2022) from OpenAI has considerably enhanced the image quality. Because DMs typically necessitates an extensive U-Net iterative denoising procedure, training becomes computationally costly. Latent Diffusion Model (LDM) (Rombach et al., 2022) and Imagen (Saharia et al., 2022) have been used to solve this issue. To increase efficiency, LDM use an autoencoder to first compress the image input, then use the produced latent space to train the DM. Our technique expands on the LDM, but maintains the original weights of the LDM, adding video perception through extra trainable temporal modules and training on video-text datasets to inherit distinct visual notions at lower computation.

2.2 Text-to-Video Generation

Autoregressive Transformer. With the development of large-scale pre-trained Transformer models in text (GPT-3) (Brown et al., 2020) and images (DALL-E) (Reddy et al., 2021), Text-to-Video (T2V) generation begin to adopt the Transformer architecture. Phenaki (Villegas et al., 2022), NÜWA (Wu et al., 2022), VideoGPT (Yan et al., 2021) and CogVideo (Hong et al., 2022) are proposed for transformer-based frameworks. A hybrid technique is proposed by TATS (Ge et al., 2022), which combines a time-sensitive transformer module for sequential frame generation with a VQGAN for image generation. Of all the transformer-based frameworks, Phenaki is one of the most intriguing. because, given a

series of signals, it may create films that are arbitrary in length (i.e., plot). NuWA-Infinity (Liang et al., 2022) presents a dual autoregressive (autoregressive over autoregressive) generation mechanism that can produce long-form, high-definition films by synthesising pictures and videos based on text input.

Two open-source models based on Transformers are VideoGPT and Cogvideo. Among these, Cogvideo's creator use the earlier T2I model. The pre-trained T2I model, according to the author, has already mastered the capacity to generate text from visual. Secondly, in order to improve text and video clip alignment, the authors suggest a hierarchical multi-frame training method. Cogvideo is paying attention to two things at once. To enable the model to master the reasoning ability of space and time, there are two types of attention: temporal attention and spatial attention, which are referred to as attention-plus and attention-base, respectively. The writers begin by comparing machine evaluations in the evaluation section. However, Cogvideo's effect is inferior than TATS, indicating that this model's influence is not very strong. In order to make their results more visually appealing, the authors also employed manual assessment. In order to assess the efficacy of Cogvideo and a few other open-source models, the author recruited ninety individuals. The performance is not very excellent, even though this article claims to be the first open source T2V transformer model. The resolution of the video the model creates will be quite poor because the model is vast, which has 9.4 billion parameters.

Diffusion Models. The fundamental characteristic of recent research trends in T2V generation models is that they use diffusion-based architectures. Diffusion models have been incredibly successful in producing hyper-realistic, varied, and contextually rich images, which spark interest in applying diffusion models to other areas, including audio, 3D, and, more recently, video. Though a lot of progress has been achieved in T2I, T2V is lagging behind because of resource-intensive training, the intrinsic complexity of modelling temporal consistency, and a lack of large, high-quality paired text-video data. Video Diffusion Model (VDM) (Ho et al., 2022c), which introduces the diffusion model to the video domain for the first time, leading the way in this wave of models. However, the resultant video has a low quality because the VDM is modelled at a low resolution. Imagen Video (Ho et al., 2022a) proposes a cascade approach to continually raise the video's quality in order to provide high-resolution generation. Make-A-Video (Singer et al., 2022) adds spatio-temporal information after producing high-resolution images from text. Additionally, a unique masked frame interpolation and extrapolation network is developed. MagicVideo (Zhou et al., 2022) synthesises videos in a low-dimensional latent space to reduce training overhead, exploring a more effective method of producing videos. Diffusion over Diffusion architecture is used by NUWA-XL (Yin et al., 2023) to generate ultra-long videos.

Latent Diffusion Models. In order to generate vidoes of excellent quality while lowering the cost of video creation, implementations utilising pre-trained LDM (Rombach et al., 2022) have been the focus of several following methods (Singer et al., 2022; He et al., 2022; Hong et al., 2022). The capabilities of T2I models are transferred to T2V in these researches. Text2video-zero (Khachatryan et al., 2023) propose a method for generating videos without the training. It contains a manual pseudo motion dynamics in pre-trained T2I models to generate short videos. Nevertheless, the videos produced in this manner suffer from poor quality and temporal inconsistency. ControlVideo (Zhang et al., 2023) is a depth-conditioned T2V method, which is based on pre-trained T2I models. It is flawed by an inherited image model that lacks design, which results in a significant text-video mismatch. As a result, a lot of research continues to concentrate on improving training. A temporal module addition approach to the pre-trained LDM is proposed by Make-A-Video (Singer et al., 2022). This allows T2I to pick up the skill of creating videos. But this approach has to be combined with a depth map. Similar methods are used by Latent-Shift (An et al., 2023), which generates video without requiring a depth map. The similarity method to us is Video LDM (Blattmann et al., 2023). Both of us add temporal modules and the weights of the initial pre-trained LDM are frozen. We extend the T2I model to the video generation model, which is particularly efficient as just a tiny portion of the training parameters are required, similar to how Video LDM works. Additionally, Video LDM trains the decoder portion of the LDM and has a frame interpolation model after the video generation model.

However, the aforementioned methods are still computationally demanding even if they produce videos using pre-trained LDM. This is due to the fact that all prior research has relied solely on 3D convolution and 3D attention, which is an excessive amount of parameters for video generation. In order to strike a compromise between pre-trained LDM and training-free methods, our method focuses on leveraging pre-trained LDM to obtain low parameter counts for video generation. This is accomplished by altering the temporal module's structural design.

CHAPTER 3

Method

Our T2V generation method is based on Latent Diffusion Model (LDM) (Rombach et al., 2022). It achieves temporal consistency video through two temporal modules. We will first elaborate the preliminary of Diffusion Models and LDMs in Section 3.1. The method pipeline will be elaborated in Section 3.2. Finally, we will introduce our proposed 1D temporal convolution and Temporal First-Adjacent Attention (TFAA) in Section 3.3.

3.1 Preliminaries

3.1.1 Diffusion Models (DMs)

Diffusion Model is a generative model. The DM is divided into a forward process and an reverse process. The inverse procedure reconstructs the picture by denoising it, while the forward process progressively adds noise to the image pixel by pixel until the image fulfils Gaussian noise. Given real images $x_0 \sim q(x)$, diffusion forward process adds Gaussian noise to them by T times cumulatively to get $x_1, x_2, \cdots, x_T$. Here, a set of hyperparameters for the Gaussian distribution's variance must be provided $\{\beta_t \in (0,1)\}_{t-1}^{T}$. The forward process can be regarded as a Markov process because each moment t is related to only moment $t-1$:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t \mathbf{I}) \tag{3.1}$$

Using the reparameterization technique, the probability distribution of x_t can be computed based on x_0:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\hat{\alpha}_t} \cdot x_0, (1-\hat{\alpha}_t) \cdot \mathbf{I}) \tag{3.2}$$

where $\alpha_t = 1 - \beta_t$, $\hat{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. As t rises, this process moves x_t closer and closer to pure noise. When $T \to \infty$, x is fully Gaussian noise. And in reality β_t is increasing as t increases, i.e. $\beta_1 <$

$\beta_2 < \cdots < \beta_T$. The denoising inference process of diffusion is the reverse process if the forward process is the process of introducing noise. We can recover the original map distribution x_0 from the whole standard Gaussian distribution if we can gradually obtain the reversed distribution $q(x_{t-1} \mid x_t)$. The function $x_T \sim \mathcal{N}(0, \mathbf{I})$. However, we can not simply infer $q(x_{t-1} \mid x_t)$. But if we know x_0, it is possible to get $q(x_{t-1} \mid x_t, x_0)$ as by the Bayesian formula:

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t \mathbf{I}) \tag{3.3}$$

The optimization objective of the DM is:

$$\mathcal{L}_{simple} = \mathbb{E}_{t,x_0,\epsilon}[\|\epsilon - \epsilon_\theta(\sqrt{\hat{\alpha}_t} \cdot x_0 + \epsilon\sqrt{1-\hat{\alpha}_t}, t)\|_2^2] \tag{3.4}$$

The expectation of the above equation is derived for data, noise and time, so the actual calculation of the loss requires sampling of data, noise and time. Since there are no input signals, the generated data is unconstrained and there is no control over the generated results. Introducing conditions can bias the generated data towards the desired result. There are many ways to introduce conditions. For example, for an image generation task, a classifier can be introduced to control the DM, and its gradient can be applied to control the image generation to favor a particular semantics, so that the model can generate the appropriate image given the label. Input images or text can also be used to guide image generation. The optimization objective of the conditional DM is:

$$\min_\theta E_{t,x_0,c,\epsilon}[\|\epsilon - \epsilon_\theta(x_t, t, c)\|_2^2] \tag{3.5}$$

where condition c is an index (discrete finite condition such as finite category condition) or continuous embedding (such as text condition).

3.1.2 Latent Diffusion Models (LDMs)

By constructing the Latent Diffusion Model (Rombach et al., 2022), which solves the resource consumption and accuracy limitation brought by the previous DM in high-dimensional features, state-of-the-art is achieved in multiple classes of downstream tasks. The LDM first maps the original features of the original 3D image $H * W * 3$ to the latent space by encoder $\mathcal{E}$. Encoder $\mathcal{E}$ compresses the image at a rate of $f = H/h = W/w$, determining f to be an integer multiple of 2. Based on compression, LDM already has the ability to perform feature operations in low dimensions. And combining the aforementioned compression model and DM, we can combine the above two to reduce the amount of computation

technically and make full use of the implicit features of the image. The optimized equation is:

$$\mathcal{L}_{LDM} := \mathbb{E}_{\varepsilon(x),\epsilon\sim\mathcal{N}(0,\mathbf{I}),t}[\|\epsilon - \epsilon_\theta(z_t, t)\|_2^2] \tag{3.6}$$

The biggest difference with the DM is that the feature part combines the features after the perceptual compression model. In many scenarios, generating models requires inputs with conditions, such as text, images, etc. Therefore, LDM has taken this capability into consideration and added the attention module to the U-net generation section:

$$Q = W_Q^{(i)} \cdot \varphi_i(z_t), K = W_K^{(i)} \cdot \tau_\theta(y), V = W_V^{(i)} \cdot \tau_\theta(y) \tag{3.7}$$

And finally the above LDM loss function is optimized into a loss function with a conditional mechanism:

$$\mathcal{L}_{LDM} := \mathbb{E}_{\varepsilon(x),y,\epsilon\sim\mathcal{N}(0,\mathbf{I}),t}[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2] \tag{3.8}$$

Compared to DM, LDM reduces the image size and memory consumption through latent space.

3.2 LDMs for T2V Generation

Our method is based on a pre-trained LDM. This is because pre-trained LDM have excellent spatial modeling capabilities. We want to turn it into a video generation model but face a challenge. Although LDM can generate high-resolution images, it cannot generate videos with temporal consistency. To address this issue, we have introduced two temporal modules. These include a 1D convolution layer to learn temporal correlation and Temporal First-Adjacent Attention (TFAA) to achieve temporal consistency. These two modules can turn LDM into a video generation model with temporal consistency. More details will be explained in Section 3.3. During the training process, we freeze all pre-trained LDM parameters to inherit its ability to generate images and only update the parameters of the two added temporal modules. In the inference process, we only need a prompt description to generate the corresponding video. The pipeline we proposed is described in detail in Figure 3.1.

Specifically, the input frame x is transformed by the pre-trained encoder $\mathcal{E}$ into the latent encoding $z_0 \in R^{B\times C\times F\times H\times W}$, where B is the batch size, C is channel of the latent dimension, F is the number of the input frames, H is the latent height shape and W is the latent width shape. After that, the latent encoding is transformed into the noise z_T through the diffusion process. z_T has the same shape as z_0. z_T is gradually denoised by the U-net in T rounds and transformed into the denoised latent encoding $\tilde{z}_0$. At the same time, the input prompt is transformed into the latent encoding z_T through the cross attention of

the pre-trained text encoder and learns to associate with the latent encoding z_T. In the training process, the latent encoding $\tilde{z}_0$ is optimized by calculating $MSELoss$ with z_0 for the model. The backbone process of our pipeline is trained using the same noise schedule as in the forward process. But the backbone process only updates the parameters of the two temporal modules ϵ_ϕ we added. The original weights ϵ_θ are frozen. The pipeline's optimization goal is:

$$\arg\min_{\phi} \mathbb{E}_{\mathcal{E}(\mathbf{x}),\epsilon\sim\mathcal{N}(0,\mathbf{I}),t}[\|\epsilon - \epsilon_{\theta,\phi}(\mathbf{z}_t;\mathbf{c},t)\|_2^2] \tag{3.9}$$

where $\mathbf{z}_t$ is the diffused latent feature $\mathbf{z} = \mathcal{E}(\mathbf{x})$ at timestep t.

In the inference process, the prompt $\mathbf{c}$ in video generation model serves as a generative condition for the diffusion process. The pipeline first generates the random noise $\mathbf{z}_T \in \mathbb{R}^{L\times C\times H\times W}$ that conforms to a Gaussian distribution. Then the pre-trained decoder $\mathcal{D}$ reconstructs the denoised latent feature $\mathbf{z}_0$ to a video.

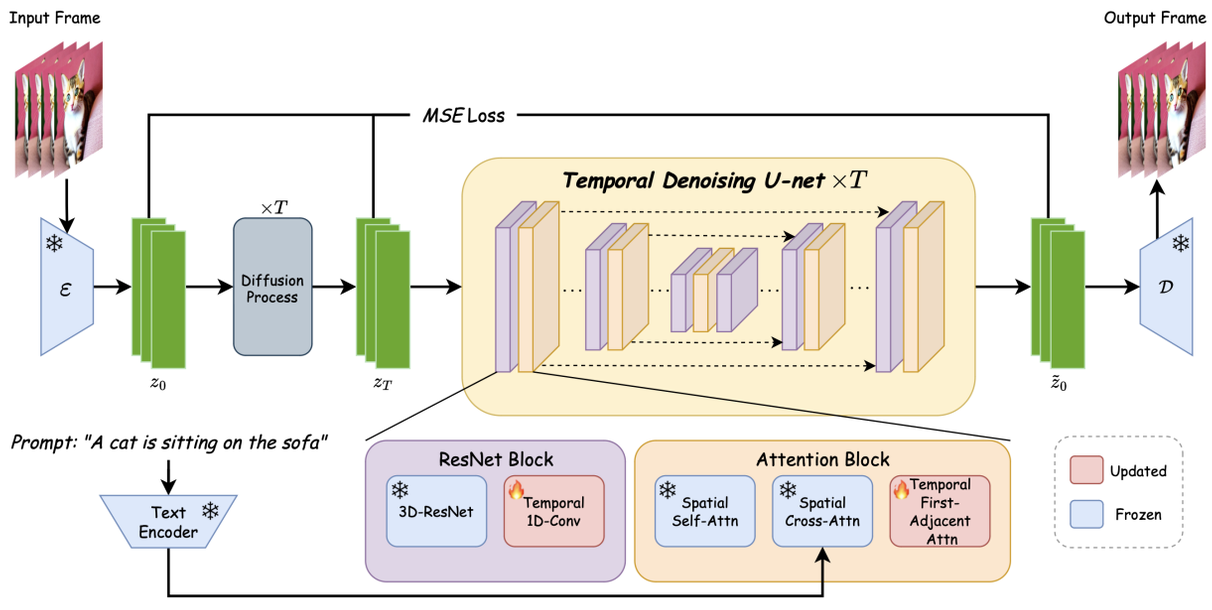


FIGURE 3.1: The pipeline of our method. The input frame is transformed into latent space through a pre-trained encoder $\mathcal{E}$ in LDM (Rombach et al., 2022). These noisy images will enter the U-Net to denoise. The $MSELoss$ is calculated to optimize the model. During training, only red modules are trainable and blue modules are frozen.

3.3 Temporal Modules

3.3.1 1D Temporal Convolution

Previously, T2V generation methods (Blattmann et al., 2023; Xing et al., 2024a; Guo et al., 2023; Xing et al., 2024b; An et al., 2023) based on LDM typically added an additional 3D convolution layer to learn temporal information for temporal consistency. But this faces a significant parameter quantity issue. Because we need to add a 3D convolution layer to each block in LDM. For a video, this will consume a lot of computing resources. Therefore, we consider that 2D convolution layers already exist in LDM to achieve high-quality spatial modeling. We only need to learn temporal modeling, which is enough. That is to say, we only need a 1D convolution layer to complete the functions of other 3D convolution layers.

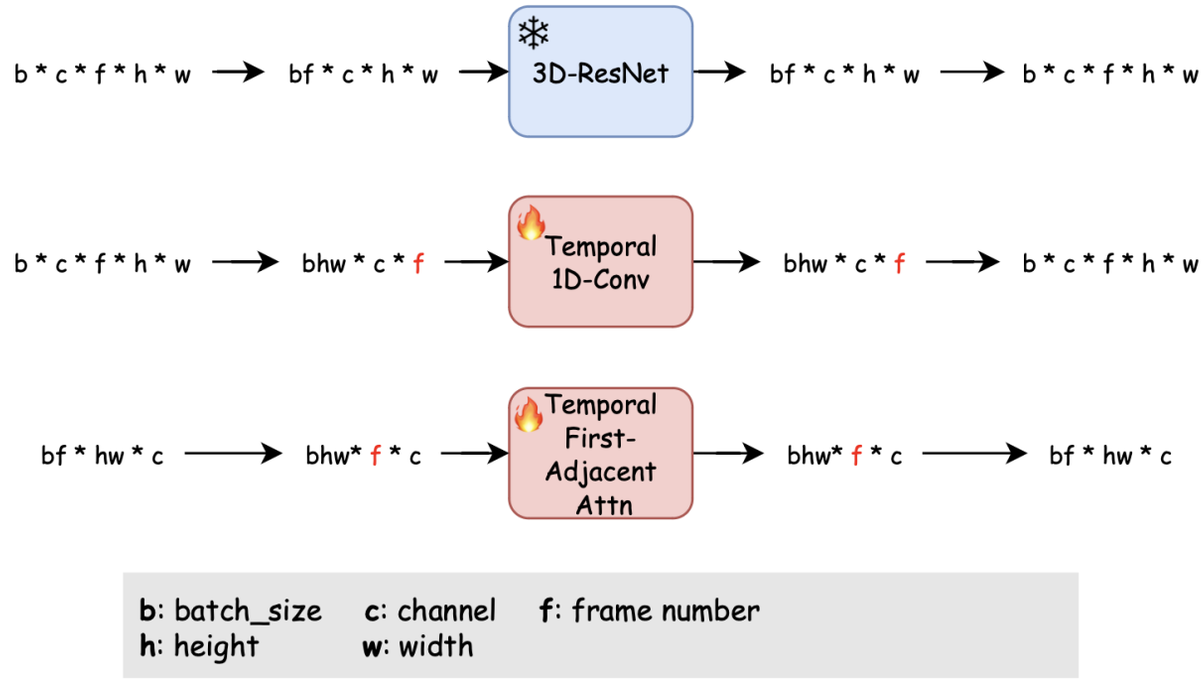


FIGURE 3.2: The dimension of the significant layers in pipeline. $3D\text{-}ResNet$ is the pre-trained 2D convolution layer in LDM. The number of frames should be times with batch size together. Therefore, the $3D\text{-}ResNet$ can be regarded as the original $2D\text{-}ResNet$ in LDM. The $Temporal\ 1D\text{-}Conv$ is followed with the $3D\text{-}ResNet$ to process the latent feature in frame dimension. The $Temporal\ First\text{-}Adjacent\ Attn$ process the attention in frame dimension.

As shown in Figure 3.2 , the input frame goes through the pre-trained encoder ε and diffusion process into latent features z_T. For each t in T, the shape of z_t is $z_t \in R^{b\times c\times f\times h\times w}$. Since the pre-trained $3D\text{-}Resnet$ in the LDM is a 2D convolution layer, we need to multiply f and b into one dimension. z_t is then transformed into a latent feature of in the shape of $z_t \in R^{bf\times c\times h\times w}$, where bf denotes that this $3D\text{-}Resnet$ learns spatial modeling one by one in the order of frames. After learning the spatial information, the latent feature z_t needs to learn the temporal consistency by our proposed $Temporal\ 1D\text{-}Conv$ layer. Therefore we keep the frame dimension f of latent feature and combine h and w with batch b into one dimension, where the shape of z_t is $z_t \in R^{bhw\times c\times f}$. The $Temporal\ 1D\text{-}Conv$ layer can only process information between different frames to achieve temporal consistency. In summary, for each $Temporal\ 1D\text{-}Conv$ layer, we reshape the latent features of the input frame as follows:

$$\begin{aligned}
\mathbf{z}' &\leftarrow \texttt{rearrange}(\mathbf{z},\ \texttt{b c f h w} \rightarrow (\texttt{b h w})\ \texttt{c f}) \\
\mathbf{z}' &\leftarrow Temporal\ 1D\text{-}Conv(\mathbf{z}') \\
\mathbf{z}' &\leftarrow \texttt{rearrange}(\mathbf{z}',\ (\texttt{b h w})\ \texttt{c f} \rightarrow \texttt{b c f h w})
\end{aligned} \tag{3.10}$$

3.3.2 Temporal First-Adjacent Attention

As shown in Figure 3.1, our $Temporal\ First\text{-}Adjacent\ Attention$ (TFAA) first adjusts the dimensions of the input latent features. By transforming the dimension, out model can learn the temporal relationship of the input frames. According to Figure 3.2, it can be seen that since TFAA is followed by a $Spatial\ Cross\text{-}Attn$. $Spatial\ Cross\text{-}Attn$ outputs the shape of latent features z_t as $z_t \in R^{bf\times hw\times c}$. This is because for the other attentions in the pipeline, they learn the spatial information. b and f are combined into one dimension, and h and w are combined into one dimension. This allows the spatial modeling of hw to be learned frame by frame. But for temporal attention, we need to combine hw and b into one dimension, and then f alone as a dimension into latent feature z_t of shape $z_t \in R^{bhw\times f\times c}$. In that case temporal attention learns information about hw for all frames f. That is, for each TFAA we have:

$$\begin{aligned}
\mathbf{z}' &\leftarrow \texttt{rearrange}(\mathbf{z},\ (\texttt{b f})\ (\texttt{h w})\ \texttt{c} \rightarrow (\texttt{b h w})\ \texttt{f c}) \\
\mathbf{z}' &\leftarrow Temporal\ First\text{-}Adjacent\ Attn(\mathbf{z}') \\
\mathbf{z}' &\leftarrow \texttt{rearrange}(\mathbf{z}',\ (\texttt{b h w})\ \texttt{f c} \rightarrow (\texttt{b f})\ (\texttt{h w})\ \texttt{c})
\end{aligned} \tag{3.11}$$

Our approach to transforming latent features above is the same as the similar methods mentioned before (Blattmann et al., 2023; Xing et al., 2024a; Guo et al., 2023; Xing et al., 2024b; An et al., 2023). However, in the previous method, self-attention (Vaswani et al., 2017) computes all frames of the input

frame. As shown in Figure 3.3, on the left is the temporal attention mechanism used by the previous method. For each frame time i the temporal attention A_i has $A_i = Attention(v_i, v_{i-1}, \cdots, v_1)$. This is because for each latent feature z_t, the attention will compute all hw in the dimension f. This is a very large computational overhead for temporal attention. Specifically, if the input frames have a total of F frames, and the sequence length of the latent features of each input frame is L. Then the computational complexity of temporal attention in the previous method is $\mathcal{O}((FL)^2)$, which is a significant computational cost for long videos. On the other hand, when this method generates the current frame based on previous frames, some previous frames may not be very relevant to the current frame. For example, v_i may only be related to v_{i-1}, but has no correlation with v_{i-2}. This will lead to inaccurate frame generation and video flickering.

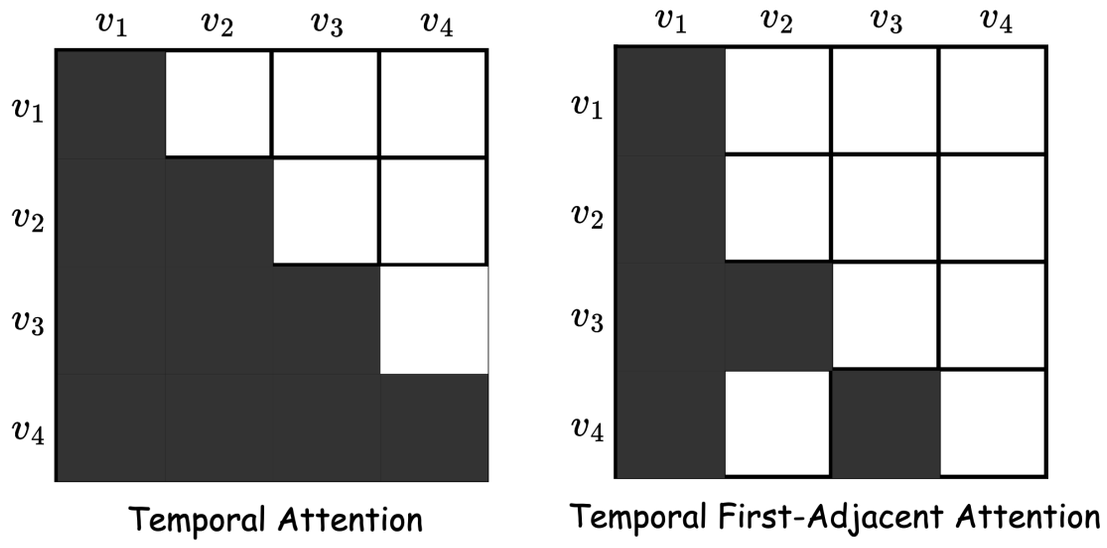


FIGURE 3.3: The comparison of temporal attention. (left): The temporal attention in previous methods, which calculate the current frame v_i with all previous frames $v_i, v_{i-1}, \cdots, v_1$. (right): Our proposed $Temporal\ First\text{-}Adjacent\ Attention$ (TFAA), which only process current frame v_i with previous frame v_{i-1} and the first frame v_1.

Specifically, the previous method can be understood as learning temporal consistency through self-attention in the frame dimension. In this self-attention, for each latent feature z_{v_i} of the input frame v_i, there are,

$$Q = W^Q \cdot z_{v_i}, K = W^K \cdot [z_{v_{i-1}}, \cdots, z_{v_1}], V = W^V \cdot [z_{v_{i-1}}, \cdots, z_{v_1}] \tag{3.12}$$

Where W represents the matrices of query Q, key K, and value V. Therefore, for the feature z_{v_i} of each video frame representing Q, the relationship between Q, K, and V can be calculated using the following

self attention formula:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}}) \cdot V \tag{3.13}$$

where QK^T is to find the degree of matching between the current frame z_{v_i} and all previous frames $[z_{v_{i-1}}, \cdots, z_{v_1}]$, that is, to identify which parts of the input sequence are most relevant to the current focus. d is the dimension of the current frame z_{v_i}, used to scale dot products and avoid getting too large values. The softmax section converts these matching scores into probability forms, where these probabilities (weights) determine the importance of each input element. Finally, these weights are used to weight the V and obtain the weighted sum output. This output is the final response of the model to the current frame z_{v_i}. In this way, this self-attention mechanism can capture the long-distance dependency relationship between the current frame and all previous frames, and generate a new representation for each frame. This representation is the weighted sum of all frames, and the weight is determined by the similarity between all frames.

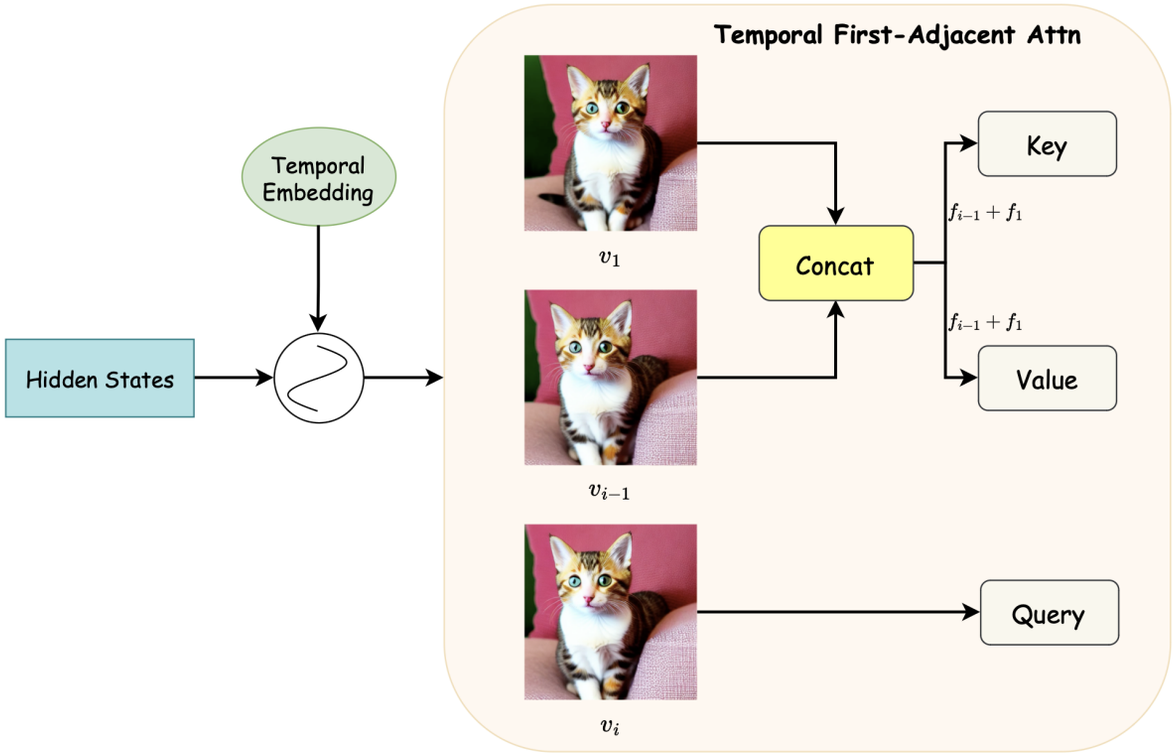


FIGURE 3.4: We start by adding a temporal embedding to the latent features. This temporal embedding is a sinusoidal positional encoding that allows the network to perceive the temporal position of the current frame. The current frame v_i is back-projected to the query Q. The key K and the value V are both the concatenation of the previous frame v_{i-1} and the first frame v_1. TFAA outputs a weighted sum of the similarities between Q and K, V.

In order to reduce such a large amount of computational consumption in the attention mechanism, we propose replacing the original attention mechanism with the TFAA on the right side in Figure 3.3. For each current frame v_i, we only calculate its correlation with the previous frame v_{i-1} and the first frame v_1. That is to say, for each frame time i, the TFAA $TFAA_i$ has $TFAA_i = TFAA(v_i, v_{i_1}, v_1)$. Therefore, the computational complexity of TFAA in our pipeline is $\mathcal{O}(2F(L)^2)$. Compared with the computational complexity of temporal attention in previous methods mentioned above, TFAA has low computational complexity. Specifically, for the latent feature z_t at each timestep t, we first add a temporal embedding to z_t. This temporal embedding is a sinusoidal position encoding that allows the network to perceive the temporal position of the current frame. Then we extract the latent feature z_{v_i} of the current frame from the latent features after adding the temporal embedding as the query Q. After that, we extract the latent feature $z_{v_{i-1}}$ of the previous frame and the latent feature z_{v_1} of the first frame. we apply concatenation f on these two latent features in the same dimension and assign the combined $f_{z_{v_{i-1}}} + f_{z_{v_1}}$ to key K and value V. Then we compute the $\mathrm{Attention}$ in the following way:

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}(\frac{Q(K^{[v_{i-1}, v_1]})^T}{\sqrt{d}}) \cdot V^{[v_{i-1}, v_1]} \tag{3.14}$$

In equation 3.14, d is the dimension of the current frame as in equation 3.13. The query Q, key K and value V represent as the following:

$$Q = W^Q \cdot z_{v_i}, K = W^K \cdot [z_{v_{i-1}}, z_{v_1}], V = W^V \cdot [z_{v_{i-1}}, z_{v_1}] \tag{3.15}$$

where, the W^Q, W^K and W^V are updated during the training process. The Figure 3.4 shows the detail and the visual information.

CHAPTER 4

Experiments

4.1 Model Architecture and Sampling

We use pre-trained weights of Stable Diffusion v1.4 to initialize our model. Because Stable Diffusion has excellent image modeling capabilities, we need it to help build our spatial layers. During the training process, we freeze the weights of the spatial layer, VAE, and text encoder from the Stable Diffusion v1.4, and only train the two temporal modules we add, which are temporal convolution layer and Temporal First-Adjacent Attention layer. In the training and inference process, we use DDIM (Song et al., 2020a) as a sampler for all experiments.

4.2 Datasets

Due to limitations in computing resources, our experiment is unable to use the commonly used WebVid-10M (Bain et al., 2021) dataset for T2V generation. Because WebVid-10M contains 10 millions text-video data, our server cannot handle such high resources. Therefore, I use two datasets to complete our experiment, namely UCF-101 (Soomro et al., 2012) and MSR-VTT (Microsoft Research Video to Text) (Xu et al., 2016). With 13320 videos spanning 101 action categories, UCF101 is an action recognition dataset for real-life action footage gathered from YouTube. Each video has a 25fps bitrate, 320×240 resolution, and an average duration of 7.21 seconds. The MSR-VTT dataset is a large-scale video dataset created by Microsoft Research Institute for video comprehension and description tasks. Its video clips come from YouTube and cover various different scenes and themes. This dataset contains 10000 video clips, each with an average duration of approximately 20 seconds.

4.3 Training Details

Our method is trained and evaluated on 2 NVIDIA-A6000 GPUs (40G). We freeze the VAE and text encoder from Stable Diffusion during the training process. For U-Net in Stable Diffusion, we only train the two temporal modules we added. Our entire model contains 1.2B parameters, while our temporal modules has 316M parameters, which means the amount of trainable parameters is 316M. For each input video, we sample its first 8 frames and set the fps to 24. In order to standardize the size of the video input, we also crop the resolution of the input video to 256×256.

4.4 Evaluation Metrics

We draw inspiration from the experiments of Video LDM (Blattmann et al., 2023) and Make Your Video (Xing et al., 2024a) and use three evaluation metrics. The first one is Fréchet Video Distance (FVD) (Unterthiner et al., 2019), which calculates the similarity between generated and real videos and captures the temporal coherence of video content as well as the quality of each frame The second one is CLIPSIM (Radford et al., 2021), which is a method used to measure video text correlation. It calculates the similarity between all frames of the output video and the corresponding editing prompts. The third is Frame Consistency (Radford et al., 2021), which evaluates the continuity between every two frames of a video. It returns the average cosine similarity between consecutive video frames and computes CLIP image embeddings on all output video frames.

CHAPTER 5

Results

5.1 Evaluation on Text-to-Video Generation

As mentioned in the Section 4.2, due to limitations in computing resources, we are unable to use the WebVid-10M dataset (Bain et al., 2021). Therefore, we train two models using UCF-101 (Soomro et al., 2012) and MSR-VTT (Xu et al., 2016), respectively. The model trained on the UCF-101 dataset is evaluated on MSR-VTT in a zero-shot manner. The model trained on the MSR-VTT dataset is evaluated on UCF-101 in a zero-shot manner.

5.1.1 Parameter Size

TABLE 5.1: **Model size comparisons**

Method	Parameters	
	Total	Trainable
Imagen Video (Ho et al., 2022a)	16.25B	16.25B
CogVideo (Hong et al., 2022)	15.5B	15.5B
Make-A-Video (Singer et al., 2022)	9.72B	9.72B
Video LDM (Blattmann et al., 2023)	4.2B	2.65B
LVDM (He et al., 2022)	1.16B	1.04B
Latent-Shift (An et al., 2023)	1.53B	880M
Make-Your-Video (Xing et al., 2024a)	2.16B	739M
AnimateDiff-v1 (Guo et al., 2023)	1.3B	417M
Ours	**1.2B**	**316M**

We compare the maount of parameters in our model with the number of trainable parameters. Except for Imagen Video (Ho et al., 2022a) and CogVideo (Hong et al., 2022), all other methods in the Table 5.1 are implemented based on diffusion models in the last year, including our baseline, Video LDM (Blattmann et al., 2023), and classic method LVDM (He et al., 2022). Our model has the minimum

number of total parameters and trainable parameters compared to all other methods. Our model has only 316M trainable parameters, which is much lower than other methods. This is because our model uses pre-trained Stable Diffusion v1.4 and our temporal attention module is a unique Temporal First-Adjacent Attention (TFAA), which greatly reduces the number of parameters. This indicates that our method can achieve temporal consistent T2V generation with minimal computational resources, greatly improving efficiency.

5.1.2 Evaluation on UCF-101

As shown in the Table 5.2, we evaluate the performance of FVD (Unterthiner et al., 2019) on the UCF-101 dataset (Soomro et al., 2012). The model we use here is trained on the MSR-VTT dataset (Xu et al., 2016). We use the classification label text of UCF-101 videos as prompts to generate all videos in a zero-shot manner, and then calculate the FVD of all videos and take the average value. Although we generate the videos in a zero-shot manner, our method's FVD value can reach 613.57, which means out method achieve good performance and consistency. We compare it with other T2V generation methods based on diffusion models, including MagicVideo (Zhou et al., 2022), LVDM (He et al., 2022), Video LDM (Blattmann et al., 2023), and Make-Your-Video (Xing et al., 2024a). The results indicate that our method can significantly outperform MagicVideo and LVDM. But it is not as good as our baseline Video LDM and Make-Your-Video, because their models are trained on WebVid-10M (Bain et al., 2021) and there are other modules that do not belong to T2V generation to improve their video performance. Although our method is not as good as baseline Video LDM, our FVD results only differ by about 60.

TABLE 5.2: **Performance comparison on UCF-101**

Method	Training Data	Zero-Shot	FVD (↓)
MoCoGAN-HD (Tian et al., 2021)	UCF-101	No	700 ± 24
CogVideo (Hong et al., 2022)	WebVid-5.4M	Yes	701.59
MagicVideo (Zhou et al., 2022)	WebVid-10M	Yes	699.00
LVDM (He et al., 2022)	WebVid-2M	Yes	641.8
Video LDM (Blattmann et al., 2023)	WebVid-10M	Yes	550.61
Text2video-zero (Khachatryan et al., 2023)	-	Yes	951.38
Make-Your-Video (Xing et al., 2024a)	WebVid-10M	Yes	367.23
Ours	MSR-VTT	Yes	**613.57**

5.1.3 Evaluation on MSR-VTT

As shown in the Table 5.3, we evaluate the performance of Frame Consistency and CLIPSIM (Radford et al., 2021) on the MSR-VTT dataset. The model we use here is trained on the UCF-101 dataset. We randomly select one prompt for each video in MSR-VTT and generate all 10k videos in a zero-shot manner. These generated videos are used to calculate Frame Consistency and CLIPSIM, and the average value is taken at the end. Although we generate the video in a zero-shot manner, our method achieves a Frame Consistency score of 0.9346 and a CLIPSIM score of 0.2902. Compared with other T2V generation methods, our method is significantly better than LVDM. Furthermore, compared to SimDA (Xing et al., 2024b), Make-Your-Video and baseline Video LDM, our method yields almost identical results with their Frame Consistency and CLIPSIM. This is still achieved when we use UCF-101 and fewer computing resources. Although other methods use larger datasets, our method also achieves equally excellent results. This indicates that the video we generate has excellent video frames consistency and text-video consistency.

TABLE 5.3: **Evaluation results on MSR-VTT**

Method	Training Data	Zero-Shot	Frame Consistency (↑)	CLIPSIM (↑)
CogVideo (Hong et al., 2022)	WebVid-5.4M	Yes	0.9064	0.2631
LVDM (He et al., 2022)	WebVid-2M	Yes	0.9317	0.2715
Text2video-zero (Khachatryan et al., 2023)	WebVid-10M	Yes	0.8148	0.2816
Video LDM (Blattmann et al., 2023)	WebVid-10M	Yes	-	0.2929
SimDA (Xing et al., 2024b)	WebVid-10M	Yes	-	0.2945
Make-Your-Video (Xing et al., 2024a)	WebVid-10M	Yes	0.9398	0.2926
Ours	UCF-101	Yes	**0.9346**	**0.2902**

5.1.4 Qualitative Results

Due to the fact that quantitative evaluation is not reliable to some extent. Therefore, we also conducted some qualitative evaluations. Our method generates videos that are compared with Text2video-zero (Khachatryan et al., 2023), Video LDM, AnimateDiff (Guo et al., 2023), and SimDA methods. All videos are generated based on a same prompt. The video results are shown in Figure 5.1. The results indicate that our video has achieved excellent results in terms of temporal consistency, text-video consistency, and video quality. Meanwhile, we also provide many of the video results we generated in Figure 5.2. But due to space limitations, we won't showcase everything here.

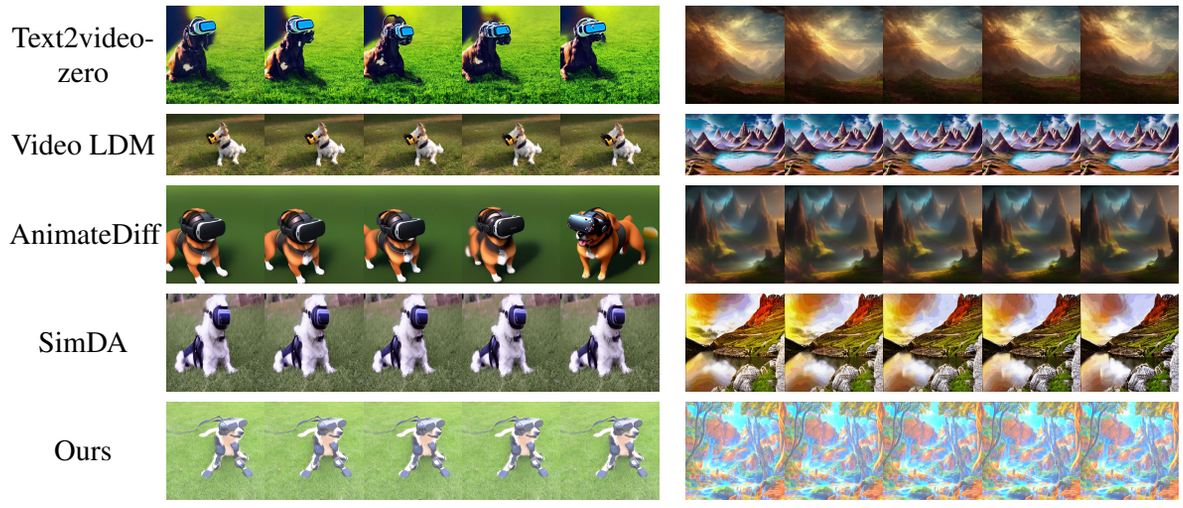


FIGURE 5.1: Text-to-Video generation results comparison with Text2video-zero (Khachatryan et al., 2023), Video LDM (Blattmann et al., 2023), AnimateDiff (Guo et al., 2023), SimDA (Xing et al., 2024b) and ours

5.2 Ablation Study

Our ablation study is conducted under the same training parameters and GPU as the training process. We used the previously mentioned FVD and CLIPSIM as evaluation metrics. Similar to the training process mentioned earlier, when evaluating FVD, we use MSR-VTT to train the model, while when evaluating CLIPSIM, we use UCF-101 to train the model. To evaluate the performance of FVD, we randomly select 1k video samples from UCF-101, and to evaluate the performance of CLIPSIM, we randomly select 1k video samples from MSR-VTT. All generated videos are implemented in a zero-shot manner. As shown in Table 5.4, ablation study involves our two temporal modules. Among them, "w/o Temporal-Attn" represents the model without additional $Temporal\ First\text{-}Adjacent\ Attention$ (TFAA), and "w/o Temporal-Conv" represents the model without $Temporal\ 1D\text{-}Conv$. Both models undergo ablation study under the same conditions.

TABLE 5.4: **Ablation Study on different modules. We report FVD on random 1k samples in UCF-101 and CLIPSIM on random 1k samples in MSR-VTT**

	FVD (↓)	CLIPSIM (↑)
w/o Temporal-Attn	1498.41	0.2668
w/o Temporal-Conv	683.24	0.2854
w/	**609.84**	**0.3015**

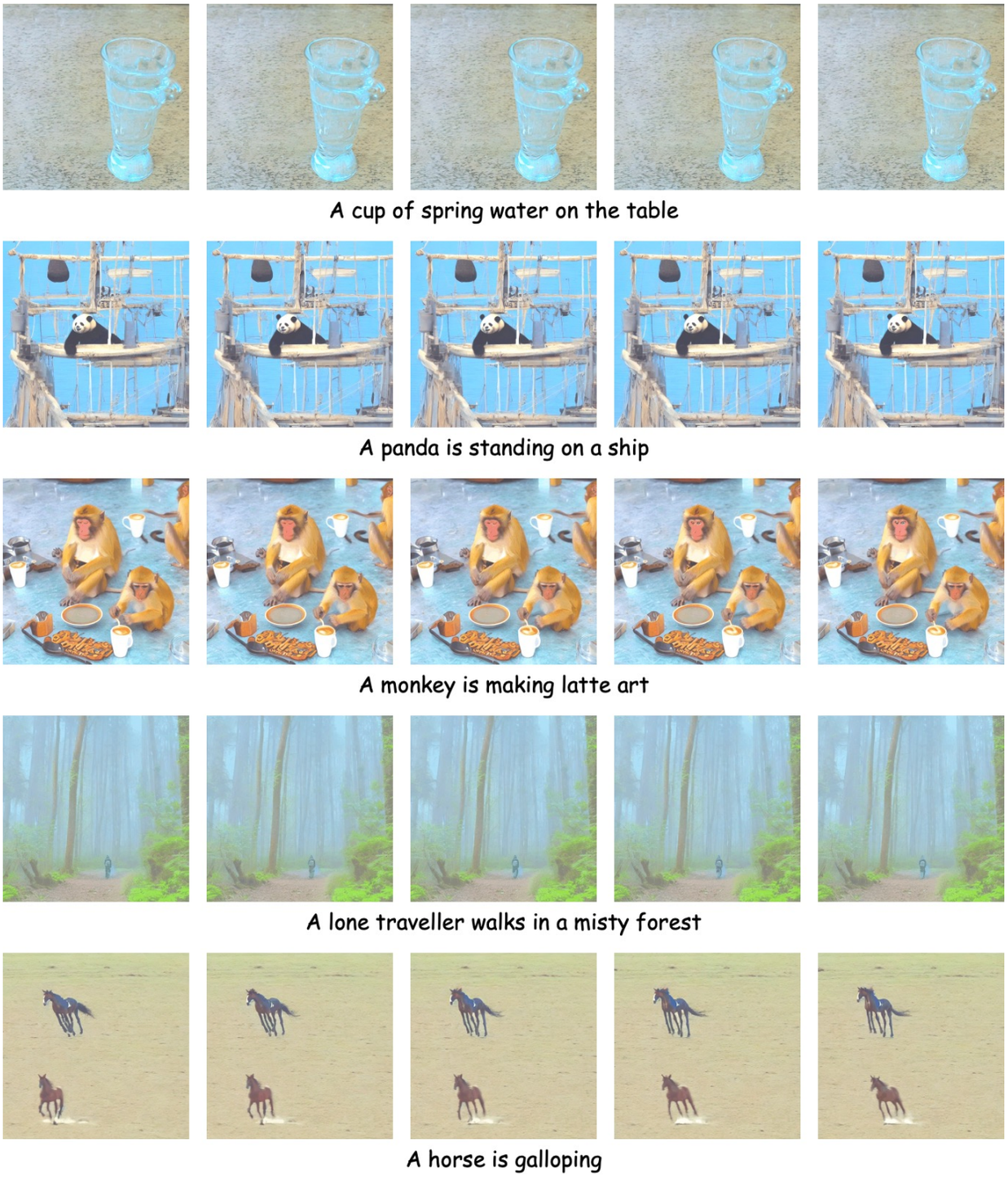


FIGURE 5.2: Qualitative results of our text-to-video generation method

From the results, we can see that the FVD of the "w/o Temporal-Attn" model, which does not include TFAA, is only 1498.41. This is because TFAA is the most important module in our method

to achieve video temporal consistency. It allows our method to learn the consistency between different frames to provide better representation for motion. In this case, the CLIPSIM is only 0.2668, which also means that TFAA provides great help for consistency between prompt and video. The FVD value of the "w/o Temporal-Conv" model without $Tmparal\ 1D\text{-}Conv$ reaches 683.24, indicating that $Tmparal\ 1D\text{-}Conv$ also affects the temporal consistency of videos to some extent, although its influence is not as significant as TFAA. This is because $Tmparal\ 1D\text{-}Conv$ is just a convolution layer for learning the relationships between video frames. In addition, the CLIPSIM of "w/o Temporal-Conv" is only 0.2854, which is 2% lower than our standard model.

Therefore, the results indicate that our temporal modules can significantly improve the temporal consistency of generated videos and the text-video consistency. We also compare the different video results under ablation study, as shown in Figure 5.3. We can see that our module can indeed generate smoother videos.

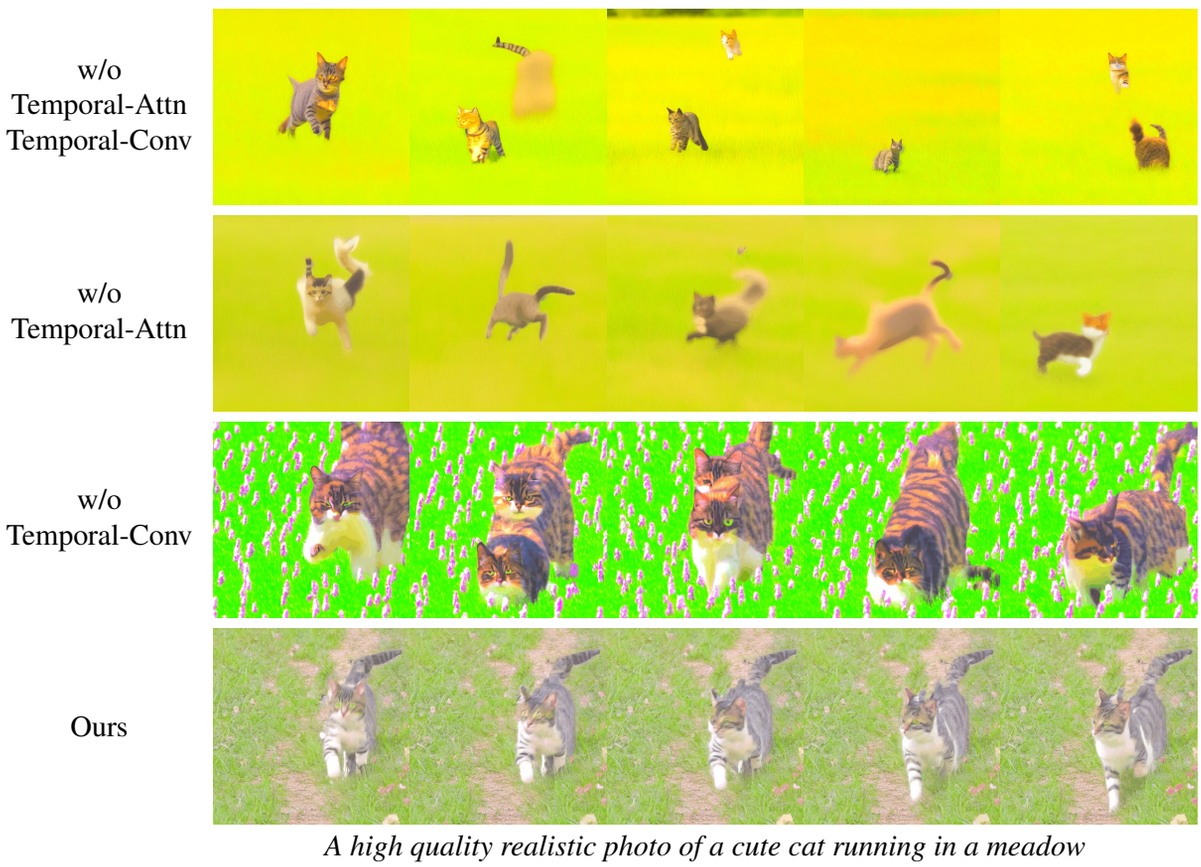


FIGURE 5.3: Ablation study of our method results. Temporal-Attn refers to the our Temporal First-Adjacent Attention (TFAA). Temporal-Conv refers to the Temproal 1D-Conv layer

CHAPTER 6

Conclusion

In this research, we propose a simple video diffusion model to achieve T2V generation. Through our proposed 1D temporal convolution layer and unique temporal attention mechanism, we extend the LDM (Rombach et al., 2022) used for image generation to a video generation model that achieves temporal consistency. We not only utilize the spatial modeling ability of LDM and the motion modeling ability of the added temporal module to generate smooth and textual videos, but also greatly reduce the training parameters and computational resources required due to our efficient temporal attention mechanism. The results indicate that our method has better T2V generation ability compared to other models (Tian et al., 2021; Hong et al., 2022; Zhou et al., 2022; He et al., 2022), while also having the smallest trainable parameters and the fastest training speed. A large number of quantitative and qualitative experiments demonstrate the effectiveness of our method.

However, there is still much future work worth paying attention to in our method. Because our method still has gaps compared to some other methods (Blattmann et al., 2023; Xing et al., 2024a), which may be caused by many factors. Firstly, due to limitations in our computing resources, our model is trained based on UCF-101 (Soomro et al., 2012) or MSR-VTT (Xu et al., 2016). Then other methods use WebVid-10M (Bain et al., 2021) as training data. Many results indicate that using WebVid-10M to train models will yield better results. Therefore, in the future, we will use WebVid-10M to train our model, which will to some extent improve the performance of our method. Additionally, although our method contains a very small number of parameters, having too few parameters may result in insufficient model training. Because our method may not be as effective in temporal modeling as other methods. Therefore, we have sufficient parameter space to add more trainable modules. We may need to add more additional temporal modules to achieve better temporal and space modeling capabilities.

Bibliography

Jie An, Songyang Zhang, Harry Yang, Sonal Gupta, Jia-Bin Huang, Jiebo Luo, and Xi Yin. 2023. Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. *arXiv preprint arXiv:2304.08477*.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

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

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. 2023. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575.

Tim Brooks, Janne Hellsten, Miika Aittala, Ting-Chun Wang, Timo Aila, Jaakko Lehtinen, Ming-Yu Liu, Alexei Efros, and Tero Karras. 2022. Generating long videos of dynamic scenes. *Advances in Neural Information Processing Systems*, 35:31769–31781.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lluis Castrejon, Nicolas Ballas, and Aaron Courville. 2019. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7608–7617.

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

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Zhida Feng, Zhenyu Zhang, Xintong Yu, Yewei Fang, Lanxin Li, Xuyi Chen, Yuxiang Lu, Jiaxiang Liu, Weichong Yin, Shikun Feng, et al. 2023. Ernie-vilg 2.0: Improving text-to-image diffusion

model with knowledge-enhanced mixture-of-denoising-experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10135–10145.

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

Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. 2022. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *European Conference on Computer Vision*, pages 102–118. Springer.

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

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

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. 2023. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*.

Sonam Gupta, Arti Keshari, and Sukhendu Das. 2022. Rv-gan: Recurrent gan for unconditional video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2024–2033.

Tanmay Gupta, Dustin Schwenk, Ali Farhadi, Derek Hoiem, and Aniruddha Kembhavi. 2018. Imagine this! scripts to compositions to videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 598–613.

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. 2022. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35:27953–27965.

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. 2022. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. 2022b. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33.

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

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. 2022. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. In *The Eleventh International Conference on Learning Representations*.

Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. 2022. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

Emmanuel Kahembwe and Subramanian Ramamoorthy. 2020. Lower dimensional kernels for video discriminators. *Neural Networks*, 132:506–520.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34:852–863.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. 2023. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. 2018. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*.

Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin. 2018. Video generation from text. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jian Liang, Chenfei Wu, Xiaowei Hu, Zhe Gan, Jianfeng Wang, Lijuan Wang, Zicheng Liu, Yuejian Fang, and Nan Duan. 2022. Nuwa-infinity: Autoregressive over autoregressive generation for infinite visual synthesis. *Advances in Neural Information Processing Systems*, 35:15420–15432.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. 2022. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer.

Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. 2020. Transformation-based adversarial video prediction on large-scale data. *arXiv preprint arXiv:2003.04035*.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. 2023. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10209–10218.

Tanya Marwah, Gaurav Mittal, and Vineeth N Balasubramanian. 2017. Attentive semantic video generation using captions. In *Proceedings of the IEEE international conference on computer vision*, pages 1426–1434.

Gaurav Mittal, Tanya Marwah, and Vineeth N Balasubramanian. 2017. Sync-draw: Automatic video generation using deep recurrent attentive architectures. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1096–1104.

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. 2022. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR.

Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. 2017. To create what you tell: Generating videos from captions. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1789–1798.

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr.

Mr D Murahari Reddy, Mr Sk Masthan Basha, Mr M Chinnaiahgari Hari, and Mr N Penchalaiah. 2021. Dall-e: Creating images from text. *UGC Care Group I Journal*, 8(14):71–75.

Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. 2016. Learning what and where to draw. *Advances in neural information processing systems*, 29.

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

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

Axel Sauer, Katja Schwarz, and Andreas Geiger. 2022. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. 2022. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*.

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

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020a. Denoising diffusion implicit models. In *International Conference on Learning Representations*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020b. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

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

Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. 2021. A good image generator is what you need for high-resolution video synthesis. In *9th International Conference on Learning Representations, ICLR 2021*.

T Unterthiner, S Van Steenkiste, K Kurach, R Marinier, M Michalski, and S Gelly. 2019. Fvd: A new metric for video generation". In *International Conference on Learning Representations Workshop(ICLRW)*.

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

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

Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. 2017. Decomposing motion and content for natural video sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017*. International Conference on Learning Representations, ICLR.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. 2022. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in neural information processing systems*,

35:23371–23385.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29.

Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. 2023. Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *arXiv preprint arXiv:2305.10874*.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. 2019. Scaling autoregressive video models. In *International Conference on Learning Representations*.

Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. 2021. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*.

Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. 2022. Nüwa: Visual synthesis pre-training for neural visual world creation. In *European conference on computer vision*, pages 720–736. Springer.

Jinbo Xing, Menghan Xia, Yuxin Liu, Yuechen Zhang, Y He, H Liu, H Chen, X Cun, X Wang, Y Shan, et al. 2024a. Make-your-video: Customized video generation using textual and structural guidance. *IEEE Transactions on Visualization and Computer Graphics*.

Zhen Xing, Qi Dai, Han Hu, Zuxuan Wu, and Yu-Gang Jiang. 2024b. Simda: Simple diffusion adapter for efficient video generation. In *CVPR*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324.

Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. 2024. Raphael: Text-to-image generation via large mixture of diffusion paths. *Advances in Neural Information Processing Systems*, 36.

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

Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. 2022. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*.

Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. 2023. Nuwa-xl: Diffusion over diffusion for extremely long video generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1309–1320.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. 2022a. Scaling autoregressive models for

content-rich text-to-image generation. *Transactions on Machine Learning Research*.

Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. 2022b. Generating videos with dynamics-aware implicit generative adversarial networks. In *10th International Conference on Learning Representations, ICLR 2022*. International Conference on Learning Representations, ICLR.

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

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. 2018. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962.

Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, XIAOPENG ZHANG, Wangmeng Zuo, and Qi Tian. 2023. Controlvideo: Training-free controllable text-to-video generation. In *The Twelfth International Conference on Learning Representations*.

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. 2022. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.