

AnimateZero: Video Diffusion Models are Zero-Shot Image Animators

Jiwen Yu^{1*}

Xiaodong Cun^{2†}

Chenyang Qi³

Yong Zhang²

Xintao Wang²

Ying Shan²

Jian Zhang^{1†}

¹ Peking University

² Tencent AI Lab

³ HKUST

<https://github.com/vvictoryuki/AnimateZero>

Generated Image



“I girl, underwater, swimsuit, ...”

Output Video

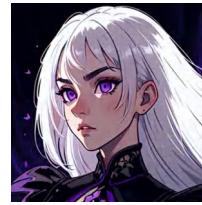
Generated Image



“I girl, black jacket, long sleeves, ...”

Output Video

Generated Image



“dark fantasy, purple eyes ...”

Output Video



“school uniform, JK, sketch ...”



“pixel art, cat ears, blonde hair ...”



“a cat head, look to one side”



“waves hit the beach”



“freckles, orange hair, glasses, ...”



“lighter, flame, candle”

Figure 1. Our proposed AnimateZero modifies the architecture of the text-to-video diffusion model, AnimateDiff [11], to achieve more controllable (e.g., control the appearance using images generated by pre-trained text-to-image models) video generation without further training. The results above demonstrate the effectiveness of AnimateZero in generating animated videos from the exactly same domains of the generated images. These personalized image domains include anime style, sketch style, pixel-art style, and realistic style. *Best viewed with Acrobat Reader. Click the video to play the animation clips. Static frames are provided in supplementary materials.*

Abstract

Large-scale text-to-video (T2V) diffusion models have great progress in recent years in terms of visual quality, motion and temporal consistency. However, the generation process is still a black box, where all attributes (e.g., ap-

pearance, motion) are learned and generated jointly without precise control ability other than rough text descriptions. Inspired by image animation which decouples the video as one specific appearance with the corresponding motion, we propose AnimateZero to unveil the pre-trained text-to-video diffusion model, i.e., AnimateDiff, and provide more precise appearance and motion control abilities for it. For appearance control, we borrow intermediate la-

* Work done during an internship at Tencent AI Lab.

† Corresponding Authors.

tents and their features from the text-to-image (T2I) generation for ensuring the generated first frame is equal to the given generated image. For temporal control, we replace the global temporal attention of the original T2V model with our proposed positional-corrected window attention to ensure other frames align with the first frame well. Empowered by the proposed methods, AnimateZero can successfully control the generating progress without further training. As a zero-shot image animator for given images, AnimateZero also enables multiple new applications, including interactive video generation and real image animation. The detailed experiments demonstrate the effectiveness of the proposed method in both T2V and related applications.

1. Introduction

Empowered by the recent development of generative priors in large-scale text-to-image (T2I) diffusion models, the video diffusion models (VDMs), especially text-to-video (T2V) diffusion models, have experienced rapid developments in terms of the resolutions [9, 14], network structures [10, 11, 13], and commercial applications [3, 6], etc.

Although VDMs are easy to use, the whole generation process is still a black box without precise control capabilities, where the users need to wait for a relatively long time to know the generated results if they have limited GPUs. Moreover, because most VDMs are trained jointly in terms of appearance and temporal aspects, it is not easy to control these two parts separately. These problems can be natively handled by generating videos by a chain of T2I and I2V (Image-to-Video). However, these two different networks, T2I and I2V model, might not be in the same domain, e.g., the T2I produces a comic image, whereas the I2V diffusion models are only trained on real-world clips. Thus, the generated results might exhibit domain bias. To this end, we are curious about the detailed generation process in the T2V generation so that we can decouple and control appearance and motion respectively and generate better videos step by step.

To achieve this goal, we are inspired by the image animation methods to consider the video as a single keyframe appearance and its corresponding movement. The keyframe can be described by the text prompt, which is a constant in the generation, and other frames utilize the knowledge of this frame for animation through the temporal modules.

Based on the above observations, we propose AnimateZero, a zero-shot method modifying the architecture of pre-trained VDMs to unveil the generation process of the pre-trained VDMs so that the appearance and motion control can be easily separated. Specifically, we have designed spatial appearance control and temporal consistency control for these two parts. Spatial appearance control involves modifying the spatial modules to insert the generated

images into the first frame of generated videos. Temporal consistency control involves modifying the motion modules to make other frames aligned with the first frame. Finally, we have achieved step-by-step video generation from T2I to I2V in a **zero-shot** manner. It is worth emphasizing that leveraging the well-established Stable Diffusion [24] community, our approach supports various personalized image domains, including but not limited to realistic style, anime style, pixel art style, and more.

Our contributions can be summarized as follows:

- We propose a novel controllable video generation method called AnimateZero, which decouples the generation progress of pre-trained VDMs, thus achieving step-by-step video generation from T2I to I2V.
- We propose spatial appearance control and temporal consistency control for AnimateZero to animate generated images in a zero-shot way. Our approach is the first to prove that the pre-trained VDMs have the potential to be zero-shot image animators.
- Experimental results highlight AnimateZero’s effectiveness in various personalized data domains. In video generation, AnimateZero surpasses AnimateDiff in similarity to the text and the T2I domain. It excels in multiple metrics compared to current I2V methods and is on par with the best method in other metrics.

2. Related Work

2.1. Text-to-Video Diffusion Models

Video Diffusion Models (VDMs) [18, 20], especially Text-to-Video Diffusion Models (T2Vs) [9, 11–13, 17, 25, 29, 30, 38, 40], have experienced rapid development recent years, making significant progress in the quality, diversity, and resolution of generated videos. Many works within these VDMs are based on tuning text-to-image diffusion models (T2Is) [24] with the addition of temporal modules. These approaches reduce the training costs of VDMs and leverage prior knowledge from the image domain. However, the tuning efforts in these works do not decouple the T2Is from the added temporal modules. Instead, they train them together, making it difficult to separate the appearance and motion control. Additionally, these methods inevitably disrupt the original T2I domain, resulting in a domain gap.

Recently, a category of VDMs that decouples T2Is and the temporal modules has emerged [4, 11]. While they provide the potential to control appearance and motion separately, they still face the challenge of disrupting the original T2I domain (demonstrated in Fig. 4). Our proposed AnimateZero is based on AnimateDiff [11].

2.2. Zero-shot Modification for Diffusion Models

Diffusion models [16, 27, 28], as representatives of large-scale vision models, have attracted considerable research

attention on how to utilize them in zero-shot or training-free manners for various downstream tasks [22, 31, 32, 36, 37]. Among these efforts, many works attempt to directly modify the model architecture to achieve new capabilities, for instance: Prompt-to-Prompt [15] modifies the cross-attention of Stable Diffusion [24] for convenient image editing; ScaleCrafter [14] modifies the convolutional kernels in the UNet of diffusion models to achieve high-quality generation at higher resolutions; MasaCtrl [8] achieves personalized image generation by sharing keys and values of the source images from the self-attention in Stable Diffusion.

Our proposed AnimateZero is also a method modifying the architecture of diffusion models, achieving zero-shot step-by-step video generation from generated images.

2.3. Image-to-Video Diffusion Models

In the realm of downstream tasks utilizing VDMs for video-related applications, there exists a category of work known as Image-to-Video Diffusion Models (I2Vs) [5, 9, 35]. The goals of these models are similar to Image Animation, but they differ in some aspects. The primary difference is that most of these methods employ an image encoder to extract semantic features from a reference image to guide video generation, without requiring the generated video to precisely include the given image as the first frame.

Recently, there have been some attempts to move towards Image Animation: publicly available tools include Gen-2 [2], Genmo [3], and Pika Labs [6]. Among them, Gen-2, as a commercial large-scale model, delivers impressive results in the realistic image domain in its November 2023 update. However, its performance in other domains, which might not have been covered in training, is still not entirely satisfactory. Genmo and Pika Labs also face the same challenge. Related research papers include SEINE [10] and LAMP [34], which are currently under submission. However, their I2V models require training and are still dependent on specific training data domains.

In comparison, our approach holds unique advantages due to its characteristic of being training-free and supporting various personalized image domains.

3. Preliminaries: AnimateDiff [11]

To simplify the experiments and hypotheses, we choose one specific video diffusion model, *i.e.*, AnimateDiff [11], as the base video model, since it only trains additional temporal layers based on a fixed text-to-image diffusion model for text-to-video generation, as shown in Fig. 2. Below, we give the details of the whole network structure of AnimateDiff and its motion modules in Section 3.1 and Section 3.2.

3.1. Architecture Overview

AnimateDiff [11] aims to learn additional temporal information on top of the pretrained large-scale text-to-image

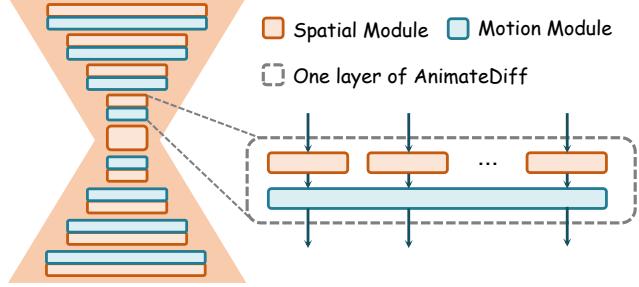


Figure 2. The architecture of the diffusion UNet in AnimateDiff [11]. It decouples the video diffusion model into two kinds of modules: the spatial module is responsible for generating appearance, and the motion module is responsible for generating motion.

model, *i.e.*, stable diffusion [24], for video generation. To achieve this, AnimateDiff decouples the video generation models into the spatial modules and motion modules individually (shown in Fig. 2). Respectively, AnimateDiff fixes the parameters of the spatial modules from the original weights of Stable Diffusion and only trains the motion modules inserted into spatial modules to generate several frames all at once. For the training dataset, the authors train motion modules of AnimateDiff on the large-scale WebVid dataset [7] with real-world videos to learn the motion prior. Interestingly, during inference, we can replace the weights of the original spatial modules (*i.e.*, the weights of the original Stable Diffusion) with various personalized checkpoints provided by the community, resulting in high visual quality videos in personalized image domains.

3.2. Details of Motion Module

The magic of AnimateDiff lies in the temporal motion modules for temporally consistent video generation. In detail, a motion module consists sequentially of a project-in linear layer, two self-attention blocks, and a project-out linear layer, respectively as shown in the middle of Fig. 3. The self-attention operates in the frame dimension, facilitating interactions between frames. Because frame-level self-attention is independent across different batches, heights, and widths, for the sake of simplicity, we omit the batch size, height, and width dimensions in our notation. We represent the input of a self-attention as $Z_{in} = \{z_1, z_2, \dots, z_f; z_i \in \mathbb{R}^{c \times 1}\}$ where f and c are numbers of frames and channels. The self-attention block first adds position embeddings $P = \{p_1, p_2, \dots, p_f; p_i \in \mathbb{R}^{c \times 1}\}$ to each input token and then projects them to queries, keys, and values, which can be described by:

$$\begin{aligned} Q &= \{q_i^i; q_i^i = W_q(z_i + p_i), 1 \leq i \leq f\}, \\ K &= \{k_i^i; k_i^i = W_k(z_i + p_i), 1 \leq i \leq f\}, \\ V &= \{v_i^i; v_i^i = W_v(z_i + p_i), 1 \leq i \leq f\}, \end{aligned} \quad (1)$$

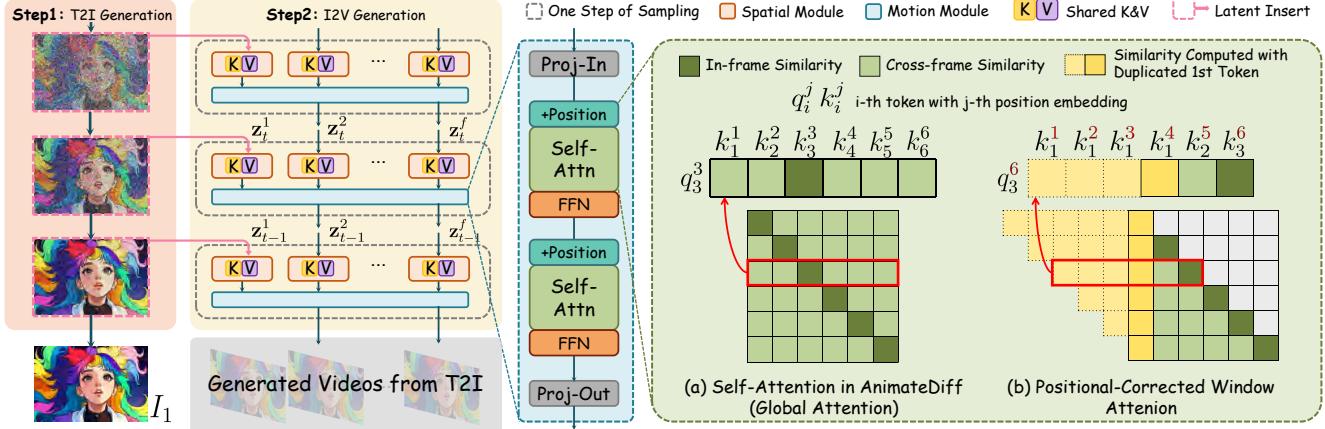


Figure 3. The overall pipeline of our proposed AnimateZero. Given spatial modules from a pre-trained T2I model [24] and its corresponding motion modules [11], we first generate a single image I_1 using the T2I model (**step1**) and then generate a video animated from this image (**step2**). The **Left** part shows the image generation process with the intermediate latents $\{z_T^1, \dots, z_0^1\}$ and our proposed Spatial Appearance Control (Sec. 4.1). Spatial Appearance Control makes modifications to the spatial modules, including the latent insertion for ensuring the first frame equal to I_1 and sharing keys and values from spatial self-attention of the first frame across other frames to align both semantics and styles. **Right** part is the Temporal Consistency Control (Sec. 4.2). We propose modifications to the original self-attention in AnimateDiff [11], which is a global attention and illustrated in (a). Our modifications include three key points (illustrated in (b)): (1) we replace global attention to window attention, which computes the i -th output token only using preceding i frames; (2) we duplicate the similarity computed with the first token to emphasize the importance of the first frame I_1 ; (3) we correct the position embeddings (marked as red in the superscripts of q and k , and the calculation of qkv is described by Eq. 2) added to input tokens to get better results.

where W_q , W_k and W_v are the linear projection parameters. Q , K and V represent queries, keys and values. The subscript “ i ” and superscript “ j ” in q_i^j indicates the addition of i -th input token z_i and j -th position embedding p_j . Here, we distinguish the serial numbers of tokens and position embeddings for the convenience of the following explanations. Finally, the calculation of output Z_{out} is:

$$Z_{out} = V \cdot \text{Softmax}(Q^\top K / \sqrt{c})^\top. \quad (2)$$

It can be observed that the temporal consistency in AnimateDiff is achieved through weighted operations of self-attention, which average all frames to get smooth results.

4. Method

Using the pre-trained AnimateDiff, our objective is to adapt it for step-by-step video generation with better visual and controllable quality. Specifically, we first generate one satisfactory image, and then utilize the intermediate latents and features of its generation process to guide the video generation. Our method consists of two parts: the spatial appearance control, discussed in Sec. 4.1, modifies the spatial modules to guarantee that the generated first frame is equal to the given image, while the temporal control, described in Sec. 4.2, modifies the motion modules to ensure temporal consistency throughout the entire generated video.

4.1. Spatial Appearance Control

We first generate an image using the same personalized T2I model in AnimateDiff, so that we can get the generated im-

age I_1 and the intermediate latents $\{z_T^1, \dots, z_t^1, \dots, z_0^1\}$ responsible for generating this image. Then, we can use these latents and features for further animation. The goal of spatial appearance control is to ensure that the first frame of the generated video is identical to I_1 . The left part of Fig. 3 illustrates the control mechanism.

Inserting Intermediate Latents. To exactly mock the generation process of image animation, for video generation, we discard the originally generated latents of the first frame in each step. Instead, we insert the intermediate latents from T2I as replacements. Notice that those intermediate latents of previous steps have not been involved in the temporal modules. This approach not only ensures that the final sampled first frame closely resembles I_1 , but also allows contents of I_1 to participate in the computation of temporal attention with other frames at each intermediate step.

Sharing K&V in Spatial Self-Attention. Relying solely on temporal attention within the motion module makes it challenging to align the semantic and style information of other frames with the first frame. Inspired by studies in personalized image generation and editing [8, 33], we make spatial modules of all frames share the same keys and values from the spatial self-attention of the first frame. The underlying implication is that each frame draws values from the same sets, implicitly ensuring similar semantic and style across frames.

4.2. Temporal Consistency Control

While we have made the first frame identical to I_1 using spatial appearance control, the motion module introduced in Sec. 3.2 does not guarantee temporal consistency. This is because the weighted operations in self-attention of motion modules are based on the computed similarity between different frames and can not automatically align other frames to a specific frame. In order to align other frames with the first frame explicitly, we propose the Positional-Corrected Window Attention to modify the original global attention (shown in the right part of Fig. 3), which will be introduced in detail below.

From Global Attention to Window Attention. First, we need to provide the formula for the self-attention calculation in the motion module, where query, key, value, and output are denoted as Q , K , V , and Z_{out} , respectively. The specific form is as follows:

$$\begin{aligned} Q &= \{q_1^1, q_2^2, \dots, q_f^f; q_i^i \in \mathbb{R}^{c \times 1}\}, & Q &\in \mathbb{R}^{c \times f}, \\ K &= \{k_1^1, k_2^2, \dots, k_f^f; k_i^i \in \mathbb{R}^{c \times 1}\}, & K &\in \mathbb{R}^{c \times f}, \\ V &= \{v_1^1, v_2^2, \dots, v_f^f; v_i^i \in \mathbb{R}^{c \times 1}\}, & V &\in \mathbb{R}^{c \times f}, \\ Z_{out} &= \{\hat{z}_1, \hat{z}_2, \dots, \hat{z}_f; \hat{z}_i \in \mathbb{R}^{c \times 1}\}, & Z_{out} &\in \mathbb{R}^{c \times f}, \end{aligned}$$

where c and f represent the numbers of channels and frames. The output \hat{z}_i for the i -th frame can be written as:

$$\hat{z}_i = V \cdot \text{Softmax}((q_i^i)^\top K / \sqrt{c})^\top. \quad (3)$$

From Eq. 3, it can be observed that the attention calculation range for each frame is global, meaning K and V include keys and values from all frames (shown in Fig. 3 (a)). Although this global design helps in averaging all frames to achieve a smooth result, it hinders the ability to align with the first frame. Therefore, our proposed improvement is the introduction of window attention (shown in Fig. 3 (b)), where the sources of keys and values for the calculation of the i -th output are limited to the preceding i frames. The specific formula can be written as:

$$\hat{z}_i = \tilde{V}_i \cdot \text{Softmax}((q_i^i)^\top \tilde{K}_i / \sqrt{c})^\top, \quad (4)$$

where $\tilde{K}_i, \tilde{V}_i \in \mathbb{R}^{c \times f}$ can be written as:

$$\tilde{K}_i = \underbrace{\{k_1^1, \dots, k_{f-i+1}^1, \dots, k_i^i\}}_{(f-i+1)}, \tilde{V}_i = \underbrace{\{v_1^1, \dots, v_{f-i+1}^1, \dots, v_i^i\}}_{(f-i+1)}. \quad (5)$$

As described in Eq. 5, we duplicate tokens from the first frame to ensure that the number of tokens in both \tilde{K}_i and \tilde{V}_i remains equal to f , emphasizing its importance during the attention computation, which further promotes alignment of other frames with the first frame.

Correct Position Embedding Makes Better Results.

Our design philosophy for the zero-shot module modification aims to ensure the operations remain unchanged from the original AnimateDiff. The local attention introduced above still has some limitations. The issue lies in the positional embeddings. Ideally, a set of keys and values should include all possible positional embeddings from p_1 to p_f . However, because the position embeddings are added before attention calculation, the i -th token only carry i -th position embedding. Therefore, \tilde{K}_i and \tilde{V}_i described in Eq. 5 include only the first i positions.

Based on this observation, we modified the mechanism for adding positional embeddings (details can be found in supplementary materials) for queries, keys, and values, so that the i -th token is added with the j -th positional embedding (i may not be equal to j). In the end, we achieved that the f tokens in \tilde{K}_i and \tilde{V}_i could carry positional embeddings from the 1-st to the f -th position, illustrated in Fig. 3 (b) and written as:

$$\begin{aligned} \tilde{K}_i &= \{k_1^1, k_1^2, \dots, k_1^{f-i+1}, k_2^{f-i+2}, \dots, k_i^f\}, \\ \tilde{V}_i &= \{v_1^1, v_1^2, \dots, v_1^{f-i+1}, v_2^{f-i+2}, \dots, v_i^f\}. \end{aligned} \quad (6)$$

Although proposed window attention has shown significant advantages over global attention in aligning other frames with the first frame, global attention tends to produce smoother results, enhancing the visual quality of the output. As we still need to increase the overall consistency via the global solution, our final solution integrates the strengths of both attentions into a Diffusion UNet. Specifically, we use a motion module with local attention in the encoder part of the UNet to align each frame with the first frame. In the decoder, we utilize a motion module with global attention to smooth all frames. We also find the time-travel sampling strategies will produce smoother results as discussed in [32, 36], which we give more additional experiments in the supplementary.

4.3. Discussion

From the proposed method, we can successfully give more control handles to the T2V generation. Also, since we find that the video diffusion model is an image animator, our method can also be considered as an image animation method for the generated image. Given the real image, we can also perform DDIM inversion [21, 26] to get the intermediate latents. Moreover, our approach, particularly the aspect related to temporal consistency control, has the potential to inspire the training of video foundation models, leading to improved training-based image-to-video models.



Figure 4. Qualitative comparison results between AnimateDiff [11] and our proposed AnimateZero. As shown in (a), (b) and (c), the videos generated by AnimateDiff are not in the same domain as the generated images. In contrast, AnimateZero is capable of maintaining consistency with the original T2I domains; In (a), (c) and (d), it is demonstrated that AnimateDiff may encounter inconsistencies between the provided text and the generated frames (highlighted in red). AnimateZero, on the other hand, performs better in this regard. *Best viewed with Acrobat Reader. Click the video to play the animation clips. Static frames are provided in supplementary materials.*

Method	Warping Error ↓	Text-Sim ↑	Domain-Sim ↑	Style-Dist ↓
AnimateDiff [11]	0.6719	0.3254	0.8081	0.3809
AnimateZero (ours)	0.6562	0.3314	0.8671	0.1666

Table 1. Quantitative comparison results between AnimateDiff [11] and our proposed AnimateZero. AnimateZero exhibits a higher similarity to the text and the original T2I domain.

5. Experiments

5.1. Implementation and Setting Details

In our experiments, spatial modules are based on Stable Diffusion V1.5 [24], and motion modules use the corresponding AnimateDiff [11] checkpoint V2. We experiment with various personalized T2I checkpoints downloaded from Civitai [1], and detailed information about these checkpoints can be found in the supplementary materials. For AnimateZero, utilizing both spatial appearance control and temporal consistency control is sufficient to achieve satisfactory results in most cases, without involving any hyper-parameters to be chosen. The length for our generated videos is 16 frames, and the video resolution is unrestricted, with a standard resolution of 512×512 .

5.2. Comparison Results

We construct a benchmark for quantitative comparison, which includes 20 prompts and 20 corresponding generated images. To achieve a comprehensive evaluation, these prompts and images include different styles (realistic and cartoon styles) and contents (characters, animals, and landscapes). Regarding evaluation metrics in Tab. 2 and Tab. 1,

we design: (1) ‘ I_1 -MSE’ uses MSE to measure whether the generated first frame matches the given image I_1 ; (2) ‘Warping Error’ [19] evaluates the temporal consistency of the generated videos; (3) ‘Text-Sim’ evaluates the similarity between the prompt and each generated frame using their features extracted by CLIP [23] Text and Image Encoders; (4) ‘Domain-Sim’ assesses the similarity between the T2I domain and the generated videos. We first use the T2I model to generate 16 images and then calculate and average the CLIP feature similarity between each of these images and each frame of the generated video; (5) ‘Style-Dist’ evaluates the style matching degree between the each generated frame and the given image I_1 , by calculating the distance between their style information which is represented by the gram matrix of the third layer features of the CLIP Image Encoder; (6) ‘User Study’, which is divided into three aspects: *Motion* evaluates the quality of the generated motion, *Appearance* assesses whether the generated appearance matches the given image I_1 , and *Subjective* evaluates the subjective quality of the generated videos. We ask 20 subjects to rank different methods in these three aspects and use the average rank number to evaluate each method.

Compared with AnimateDiff. While AnimateDiff [11] demonstrates good generalization ability on many personalized T2I models, it occasionally produces low-quality videos (shown in Fig. 4), especially on anime-style T2I models. These low-quality videos mainly manifest in two aspects: (1) the generated videos are not within the same domain as the original T2I models; (2) a decrease in text-frame

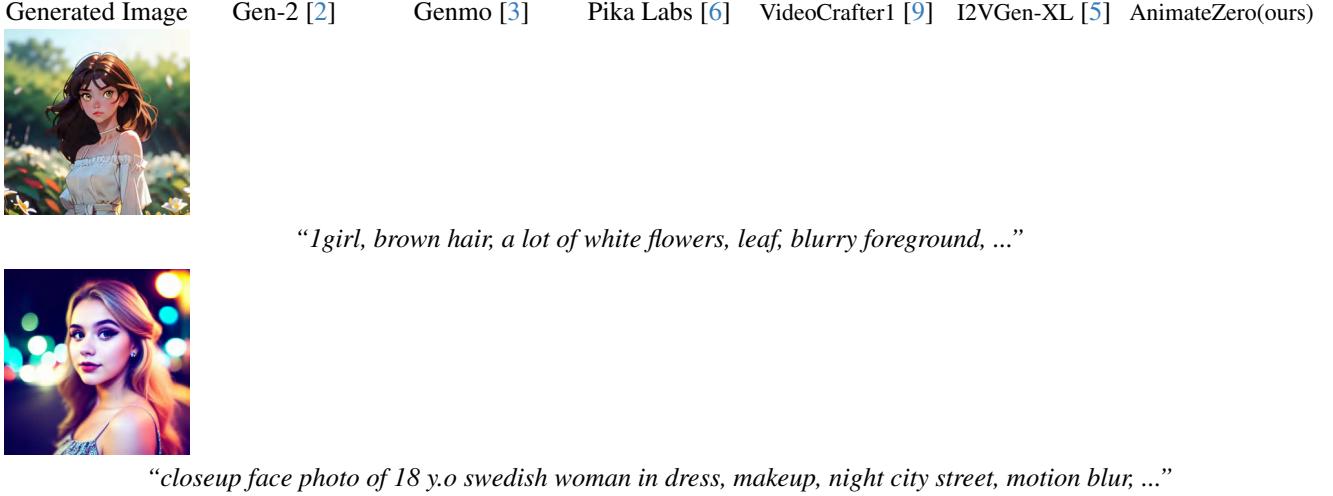


Figure 5. Qualitative comparison results between publicly available image-to-video tools and our proposed AnimateZero. *Best viewed with Acrobat Reader. Click the video to play the animation clips. Static frames are provided in supplementary materials.*

Method	Basic Metrics		CLIP Metrics			User Study		
	I_1 -MSE \downarrow	Warping Error \downarrow	Text-Sim \uparrow	Domain-Sim \uparrow	Style-Dist \downarrow	Motion \downarrow	Appearance \downarrow	Subjective \downarrow
Gen-2 [2]	59.93	0.7353	0.3282	0.7796	0.1707	3.57	2.52	2.88
Genmo [3]	90.76	0.8284	0.3184	0.7801	0.2752	2.96	3.51	3.21
Pika Labs [6]	37.68	0.6018	0.3372	0.7876	0.1275	3.71	2.18	2.84
VideoCrafter1 [9]	96.23	0.6596	0.3325	0.7598	0.2762	4.29	5.09	4.91
I2VGen-XL [5]	104.8	0.7724	0.3009	0.7272	0.4308	4.63	5.79	5.38
AnimateZero (Ours)	1.136	0.6562	0.3314	0.8671	0.1666	1.83	1.91	1.78

Table 2. Quantitative comparison results between publicly available Image-to-Video tools and our proposed AnimateZero. Our proposed AnimateZero demonstrated best performance across multiple metrics or achieved comparable results to the best methods in other metrics. The metrics for the best-performing method are highlighted in red, while those for the second-best method are highlighted in blue.

alignment in the generated videos. Surprisingly, in our experiments, we find that AnimateZero excels in both of these aspects compared to AnimateDiff, which has been demonstrated in Fig. 4. In Tab. 1, we also quantitatively evaluate AnimateDiff and AnimateZero on our benchmark at four metrics. Our proposed AnimateZero outperforms AnimateDiff in all four metrics in terms of text-frame alignment and matching degree between the generated videos and original T2I domains.

Compared with Publicly Available I2V Tools. Existing I2V methods claim to be versatile but still struggle with domain gap issues. In our experiments, we use the generated image as a condition for video creation, ensuring alignment with the T2I domain. This aims to explore AnimateZero’s advantages over existing I2V methods and highlight their limitations. We compare AnimateZero with several publicly available image-to-video tools, both closed-source (Gen-2 [2], Genmo [3], Pika Labs [6]) and open-source (VideoCrafter [9], I2VGen-XL [5]), using benchmark images and their corresponding prompts. In terms of subjective quality, as shown in Fig. 5, our proposed AnimateZero achieves performance comparable to, or even bet-

ter than, the current state-of-the-art Gen-2 and Pika Labs, standing out as the best among open-source tools. In contrast, Genmo, VideoCrafter and I2VGen-XL can only leverage the semantic information of the given generated images, failing to ensure the first frame matches the given image. Gen-2, Genmo, VideoCrafter and I2VGen-XL suffer from domain gap issues, particularly noticeable in anime-style images, whereas AnimateZero does not encounter this problem. We also conduct a comprehensive evaluation of AnimateZero and these I2V methods across all metrics in Tab. 2. It can be observed that our proposed AnimateZero achieves the best performance in certain metrics and is comparable to the best methods in other metrics. Considering that AnimateZero is a method that does not require additional training specifically for image animation, achieving the mentioned performance is highly remarkable.

5.3. Ablation Study

We conduct ablation experiments on the spatial appearance control (introduced in Sec. 4.1) and temporal consistency control (introduced in Sec. 4.2). The experimental results are shown in Fig. 6 to illustrate the role of each component in our proposed method. Firstly, Fig. 6 (a) shows the results

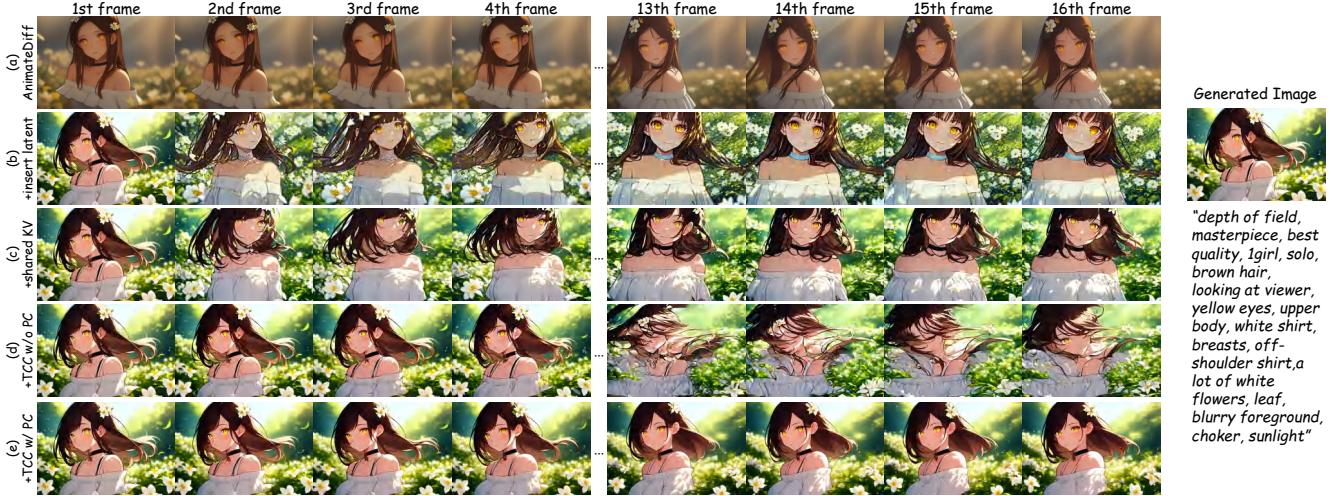


Figure 6. Demonstration for ablation study: (a) the video generated by AnimateDiff [11]; (b) +inserting intermediate latents responsible for the generation of the given image; (c) +sharing keys and values from the generation of the given image; (d) +temporal consistency control without position correction (TCC w/o PC); (e) +temporal consistency control with position correction (TCC w/ PC). To clearly illustrate the role of each component, we present static frames, while *dynamic videos are provided in the supplementary materials*.

generated by AnimateDiff with the provided text, which serves as the baseline for our ablation experiments. We will demonstrate the step-by-step process of incorporating our proposed techniques to achieve animation of the generated image. In Fig. 6 (b), we insert the intermediate latents, making the first frame almost identical to the generated image. This also implicitly controls the content and style of the other frames. However, notable differences persist in terms of style and colors when compared to the generated image. In Fig. 6 (c), we employ the strategy of sharing keys and values, further aligning the style and semantic information between the first frame and other frames. However, the spatial appearance control mentioned above cannot guarantee a seamless connection between the first frame and the rest frames. This is where our temporal consistency control (TCC) comes into play. We first attempt TCC without position correction (TCC w/o PC) in Fig. 6 (d), which ensures the temporal connection of the first several frames. However, the quality of frames towards the end of the video significantly deteriorates. This is addressed by employing TCC with position correction (TCC w/ PC) in Fig. 6 (e).

5.4. Limitations

Although our method enables the possibility of both controllable video generation and image animation, there are still some limitations. These limitations mainly stem from the constraints in motion prior within AnimateDiff [11]. AnimateDiff struggles to generate complex motions, such as sports movements or the motion of uncommon objects (demonstrated in Fig. 7). In theory, since the generated motion of AnimateZero relies on motion prior of AnimateDiff, AnimateZero is also less proficient in creating videos in the mentioned scenarios. However, we believe these limi-

AnimateDiff AnimateZero AnimateDiff AnimateZero

"Iboy, playing football, ..." *"robot, running, ..."*

Figure 7. AnimateZero is limited by the motion prior of AnimateDiff [11], and both perform poorly in complex movements. *Best viewed with Acrobat Reader. Click the video to play the animation clips. Static frames are provided in supplementary materials.*

tations can be solved with a better video foundation model with more powerful motion prior.

6. Conclusions

In this paper, we present AnimateZero, which considers video generation as an image animation problem, allowing us to modify the pre-trained video diffusion model to enable more controllability in terms of appearance and motion. To achieve this, for appearance control, we inject the generative latents into the video generation so that we can generate the video concerning the first frame. For motion control, we propose a positional corrected window attention in the motion modules to generate temporally consistent results. Experiments show the advantage of AnimateZero compared with the AnimateDiff and the general image-to-video algorithms. AnimateZero is also the first to show that video diffusion models are zero-shot image animators, which not only allows controllable video generation but also opens up possibilities for various applications like animating real images, interactive video creation, and more.

References

- [1] Civitai. <https://civitai.com/>, 2023. 6, 11
- [2] Gen-2. <https://runwayml.com/ai-magic-tools/gen-2/>, 2023. 3, 7, 17, 18
- [3] Genmo. <https://www.genmo.ai/>, 2023. 2, 3, 7, 17, 18
- [4] Hotshot-xl. <https://github.com/hotshotco/Hotshot-XL>, 2023. 2
- [5] I2vgen-xl. <https://modelscope.cn/models/damo/Image-to-Video/summary>, 2023. 3, 7, 17, 18
- [6] Pika labs. <https://www.pika.art/>, 2023. 2, 3, 7, 17, 18
- [7] Max Bain, Arsha Nagrani, GÜl Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021. 3
- [8] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactr: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3, 4
- [9] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023. 2, 3, 7, 17, 18
- [10] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. *arXiv preprint arXiv:2310.20700*, 2023. 2, 3
- [11] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 1, 2, 3, 4, 6, 8, 12, 16, 19
- [12] Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [13] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. 2022. 2
- [14] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. 2023. 2, 3
- [15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. 2022. 3
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 12, 13
- [17] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2
- [18] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 2
- [19] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *European Conference on Computer Vision*, 2018. 6
- [20] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2
- [21] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 5
- [22] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3
- [23] 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 *International Conference on Machine Learning (ICML)*. PMLR, 2021. 6
- [24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 4, 6
- [25] 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. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 2
- [26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5, 13
- [27] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2
- [28] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021. 2
- [29] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 2

- [30] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 2
- [31] Yinhuai Wang, Jiwen Yu, Runyi Yu, and Jian Zhang. Unlimited-size diffusion restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023. 3
- [32] Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *International Conference on Learning Representations (ICLR)*, 2023. 3, 5, 11, 12
- [33] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 4
- [34] Ruiqi Wu, Liangyu Chen, Tong Yang, Chunle Guo, Chongyi Li, and Xiangyu Zhang. Lamp: Learn a motion pattern for few-shot-based video generation, 2023. 3
- [35] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. 2023. 3
- [36] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. *arXiv:2303.09833*, 2023. 3, 5, 11, 12
- [37] Jiwen Yu, Xuanyu Zhang, Youmin Xu, and Jian Zhang. Cross: Diffusion model makes controllable, robust and secure image steganography. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3
- [38] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*, 2023. 2
- [39] Lvmi Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. 2023. 12
- [40] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 2

This appendix includes our supplementary materials as follows:

- Section A: Implementation details of utilized T2I checkpoints and the proposed position-enhanced window attention.
- Section B: Introduce the effect of time-travel sampling strategy.
- Section C: Introduce the extensive applications of AnimateZero.
- Section D: Provide more visual results. Dynamic videos can be found in our project page: <https://victoryuki.github.io/animatezero.github.io/>

Appendix A. Implementation Details

A.1. Utilized Personalized T2I Checkpoints

To thoroughly evaluate the performance of our approach across diverse image domains, we obtain personalized T2I checkpoints featuring various styles from Civitai [1] for assessment. Detailed information regarding these checkpoints is provided in Tab. 3.

Model Name	Model Type	Image Domain	URL
ToonYou	checkpoints	Anime	https://civitai.com/models/30240
CarDos Anime	checkpoints	Anime	https://civitai.com/models/25399
Anything V5	checkpoints	Anime	https://civitai.com/models/9409
Counterfeit V3.0	checkpoints	Anime	https://civitai.com/models/4468
Realistic Vision V5.1	checkpoints	Realistic	https://civitai.com/models/4201
Photon	checkpoints	Realistic	https://civitai.com/models/84728
helloObject	checkpoints	Realistic	https://civitai.com/models/121716

Table 3. The sources of all personalized T2I checkpoints used in our experiments.

A.2. Position Embedding Correction in Self-Attention

This subsection presents the detailed calculation process of self-attention with position embedding correction. First, we omit the batch size, height, and width dimensions in our notation, assuming that the input tokens of self-attention is $Z_{in} = \{z_1, z_2, \dots, z_f; z_i \in \mathbb{R}^{c \times 1}\}$ where c and f represent the numbers of channels and frames. In the **first** step, we add position embeddings, denoted as $P = \{p_1, p_2, \dots, p_f; p_i \in \mathbb{R}^{c \times 1}\}$, to input tokens. We perform pairwise addition of each element in P and Z_{in} , constructing a set $\{a_i^j; a_i^j = z_i + p_j, 1 \leq i \leq f, 1 \leq j \leq f\}$. In the **second** step, we construct a position-corrected pool for queries, keys and values:

$$\{q_i^j, k_i^j, v_i^j; q_i^j = W_q a_i^j, k_i^j = W_k a_i^j, v_i^j = W_v a_i^j, 1 \leq i \leq f, 1 \leq j \leq f\}, \quad (7)$$

where W_q , W_k and W_v represent the linear projection weights of queries, keys and values. In the **third** step, we obtain the output $Z_{out} = \{\hat{z}_1, \hat{z}_2, \dots, \hat{z}_f; \hat{z}_i \in \mathbb{R}^{c \times 1}\}$ by calculating the proposed window attention:

$$\hat{z}_i = \tilde{V}_i \cdot \text{Softmax}((q_i^f)^\top \tilde{K})_i / \sqrt{c}. \quad (8)$$

The used keys \tilde{K}_i and values \tilde{V}_i are all limited in first i frames, that is,

$$\tilde{K}_i = \{k_1^1, k_1^2, \dots, k_1^{f-i+1}, k_2^f, \dots, k_i^f\}, \quad \tilde{V}_i = \{v_1^1, v_1^2, \dots, v_1^{f-i+1}, v_2^f, \dots, v_i^f\}, \quad (9)$$

where the token from first frame have been copied for $f - i + 1$ times to keep the total numbers of utilized keys and values equal to f . One detail worth noting is that the superscript of the query token from the i -th frame in Eq. 8 should be consistent with the superscript of the corresponding key and value tokens from i -th frame in Eq. 9.

Appendix B. Effect of Time-Travel Sampling Strategy

As proposed in [32, 36], the time-travel sampling strategy has the ability to improve the visual quality of sampled results. In our experiments, this strategy generates smoother video according to the chosen hyperparameters. Assuming that the intermediate video latent code at t -th timestep is $\mathbf{Z}_t = \{\mathbf{z}_t^1, \mathbf{z}_t^2, \dots, \mathbf{z}_t^f\}$, where \mathbf{z}_t^i is the intermediate latent code for i -th frame. For one denoising step, the calculation formula is

$$\mathbf{Z}_{t-1} = \text{Denoiser}(\mathbf{Z}_t, \mathbf{t}_{prompt}, t), \quad (10)$$



Figure 8. Demonstrate the role of time-travel sampling strategy [32, 36]. The time-travel sampling strategy can produce smoother and more natural results. However, it should be emphasized that in most cases, AnimateZero can already obtain satisfactory results. The time-travel sampling strategy is only used in certain T2I models (such as Anything V5) or certain complex textures (such as hair). *Best viewed with Acrobat Reader. Click the video to play the animation clips. Static frames are provided in Sec. D.*

in which $Denoiser(\cdot)$ represents the denoising process and t_{prompt} is the given prompt text. The time-travel sampling strategy iteratively performs denoising operation in Eq. 10 and diffusion operation [16] in:

$$\mathbf{Z}_t = \sqrt{\alpha_t} \mathbf{Z}_{t-1} + \sqrt{1 - \alpha_t} \mathbf{N}, \mathbf{N} = \{\epsilon_1, \epsilon_2, \dots, \epsilon_f; \epsilon_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\}. \quad (11)$$

This approach aims to achieve more harmonious and natural generation results. In the context of video generation tasks, we observed that it has the ability to make the final video smoother. The number of iterations in each timestep determines the degree of smoothness in the final results, and we set the default number as 5. According to the findings in [36], it is unnecessary to apply this strategy at every timestep. Instead, we use the time-travel sampling strategy only between the 10-th timestep and the 20-th timestep. We show the comparison results before and after using time-travel sampling strategy in Fig. 8.

Appendix C. Applications

C.1. Improved Video Editing Compared to AnimateDiff [11]

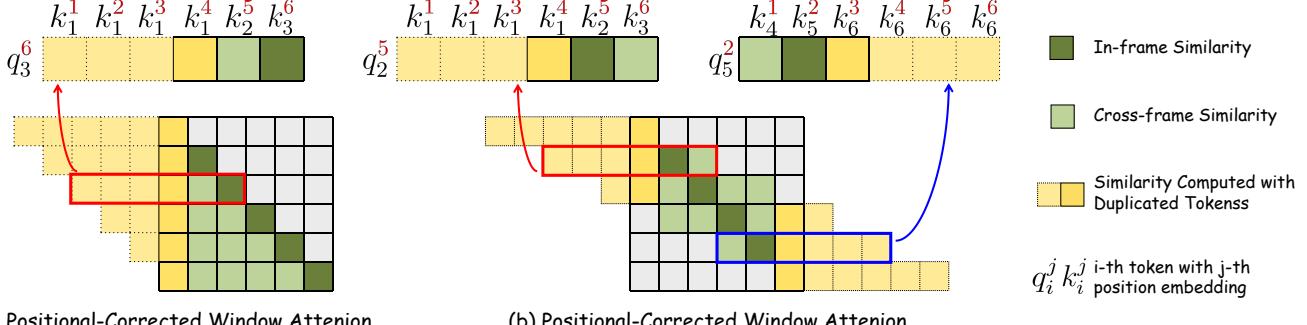
The temporal consistency of videos generated by AnimateDiff is notable, and a common use of AnimateDiff is to assist ControlNet [39] in video editing, aiming to achieve smooth editing results. The specific approach involves inputting feature information for each frame of the original video (such as extracted depth maps, edge maps, etc.) to ControlNet, thereby controlling each frame of the video generated by AnimateDiff. The primary challenge encountered in this video editing process is the inherent domain gap issue of AnimateDiff. This issue significantly degrades the subjective quality of the edited video, and the alignment degree between the text and the generated video is also substantially reduced. As demonstrated in the experimental section of the main paper, AnimateZero exhibits a significant advantage in maintaining the T2I domain compared to AnimateDiff. Therefore, we attempted to use AnimateZero to assist ControlNet in video editing. The results of the editing process showed a noticeable improvement in subjective quality and text-video matching degree compared to AnimateDiff. Additionally, AnimateZero still ensures that the generated video remains smooth and good temporal consistency. *We showcase some results of AnimateZero performing video editing in our project page.*

C.2. Frame Interpolation and Looped Video Generation

AnimateZero attempts to insert the first frame into the generated video to achieve image animation from the generated image. An extended idea is whether similar techniques can be used to insert **multiple** frames. In Fig. 9, we propose an extension to the original position-corrected window attention used by AnimateZero. This extension allows for the simultaneous insertion of both the first and last frames. The key modification involves simultaneously emphasizing the tokens corresponding to both the first and last frames, ensuring that the final generated video's first and last frames match the given images. This technique has the potential application for frame interpolation, allowing interpolation between any two generated images. Additionally, when the first and last frames are the same, it can be considered as achieving looped video generation. *Relevant results are showcased in our project page.*

C.3. Real Image Animation

AnimateZero has been demonstrated to perform image animation on generated images, but it also has the potential to handle image animation on real images. The main difference between real and generated images is the absence of readily available



(a) Positional-Corrected Window Attention
(insert the **first** frame)

(b) Positional-Corrected Window Attention
(insert the **first** and **last** frame)

Figure 9. Demonstrate the difference between (a) insertion of the **first** frame and (b) insertion of both **first** and **last** frames. The technique illustrated in (b) is the basis for achieving applications like frame interpolation and looped video generation.

intermediate latents. However, we can obtain pseudo intermediate latents through methods like DDIM Inversion [26] or by directly diffusing the clean latents [16], enabling image generation on real images. Nevertheless, for real images, the issue of domain gap is challenging to avoid. This is influenced not only by the style of the real image but also factors such as its resolution and whether it has been degraded by some degradation operators. *We showcase some results of AnimateZero performing real image animation in our project page.*

Appendix D. More Visual Results

To facilitate the accessibility of visual results for non-Adobe users, we have included *static frames* of videos in main paper (Fig. 1, 4, 5, 7) and supplementary materials (Fig. 8) in Fig. 10, 11, 12, 13, 14, 15, 16,. We also provide an *HTML file* with many generated video examples, which we highly recommend readers to check out. In addition, we have also attached the *source files* of all videos in the paper and HTML file for the convenience of readers.



“1girl, underwater, swimsuit, air bubble, looking at viewer, swimming, dappled sunlight, ...”



“1girl, black jacket, long sleeves, pink hair; hair between eyes, sitting, ...”



“dark fantasy, purple eyes, cinematic light, white hair, sharp face, hair between eyes, ...”



“school uniform, JK, sketch, long hair; clam down, looking at the camera, ...”

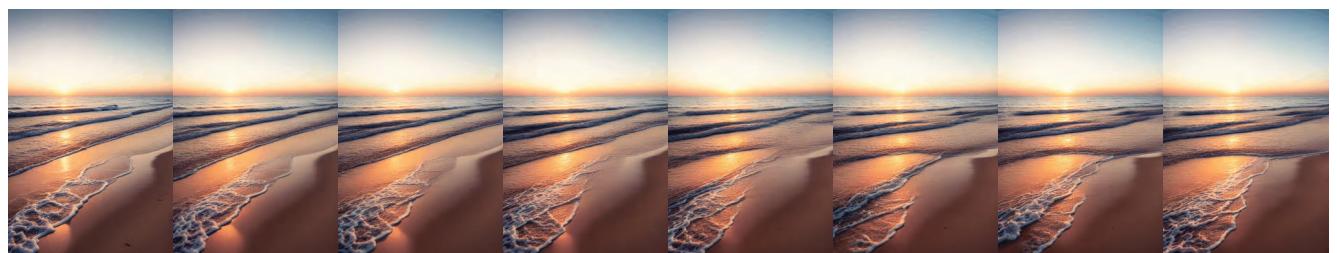


“pixel art, cat ears, blonde hair, wavy hair, portrait of cute girl, ...”

Figure 10. Static frames sequences in Fig. 1 (part 1).



"a cat head, look to one side"



"waves hit the beach"



"freckles, orange hair, glasses, ..."



"lighter, flame, candle"

Figure 11. Static frames sequences in Fig. 1(part 2).

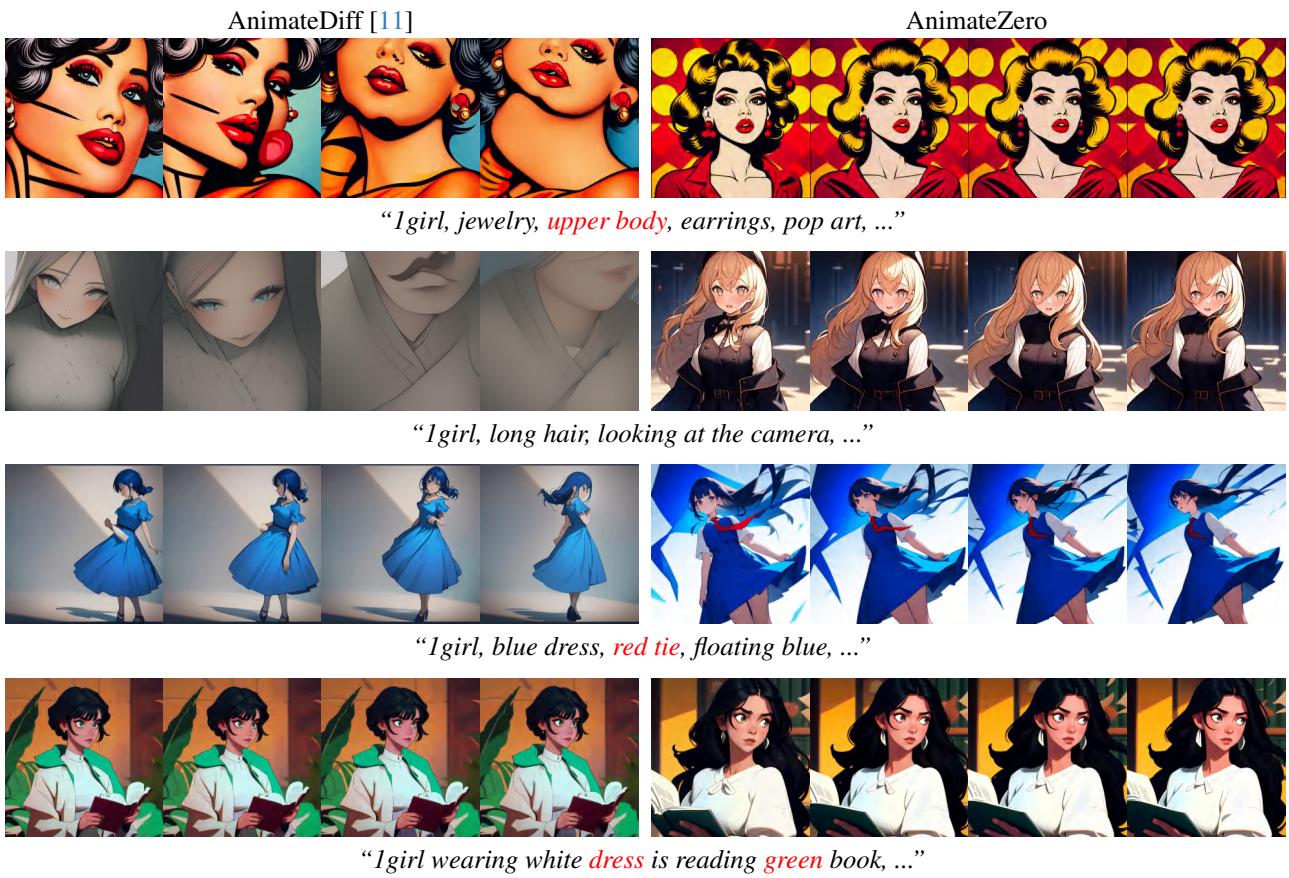
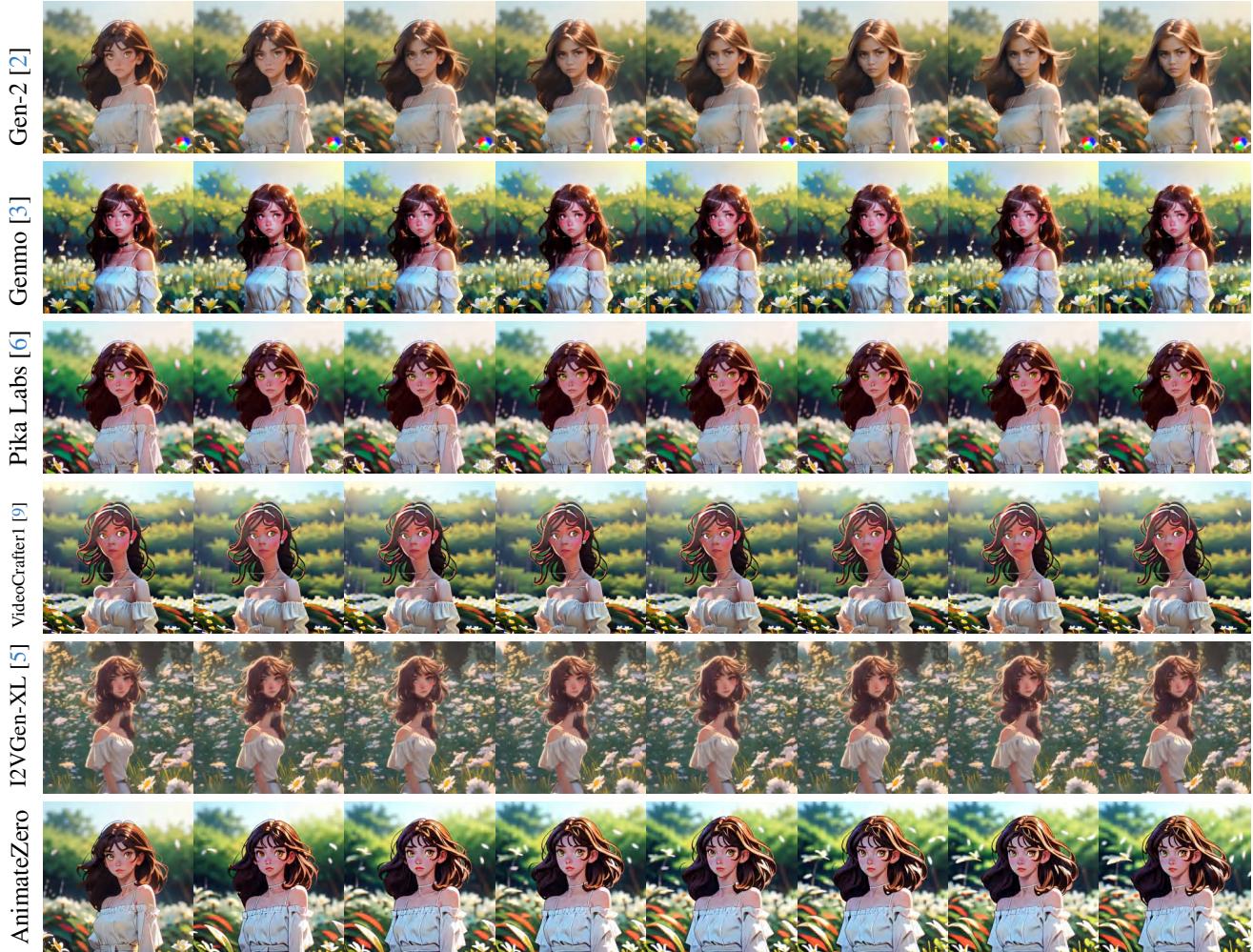
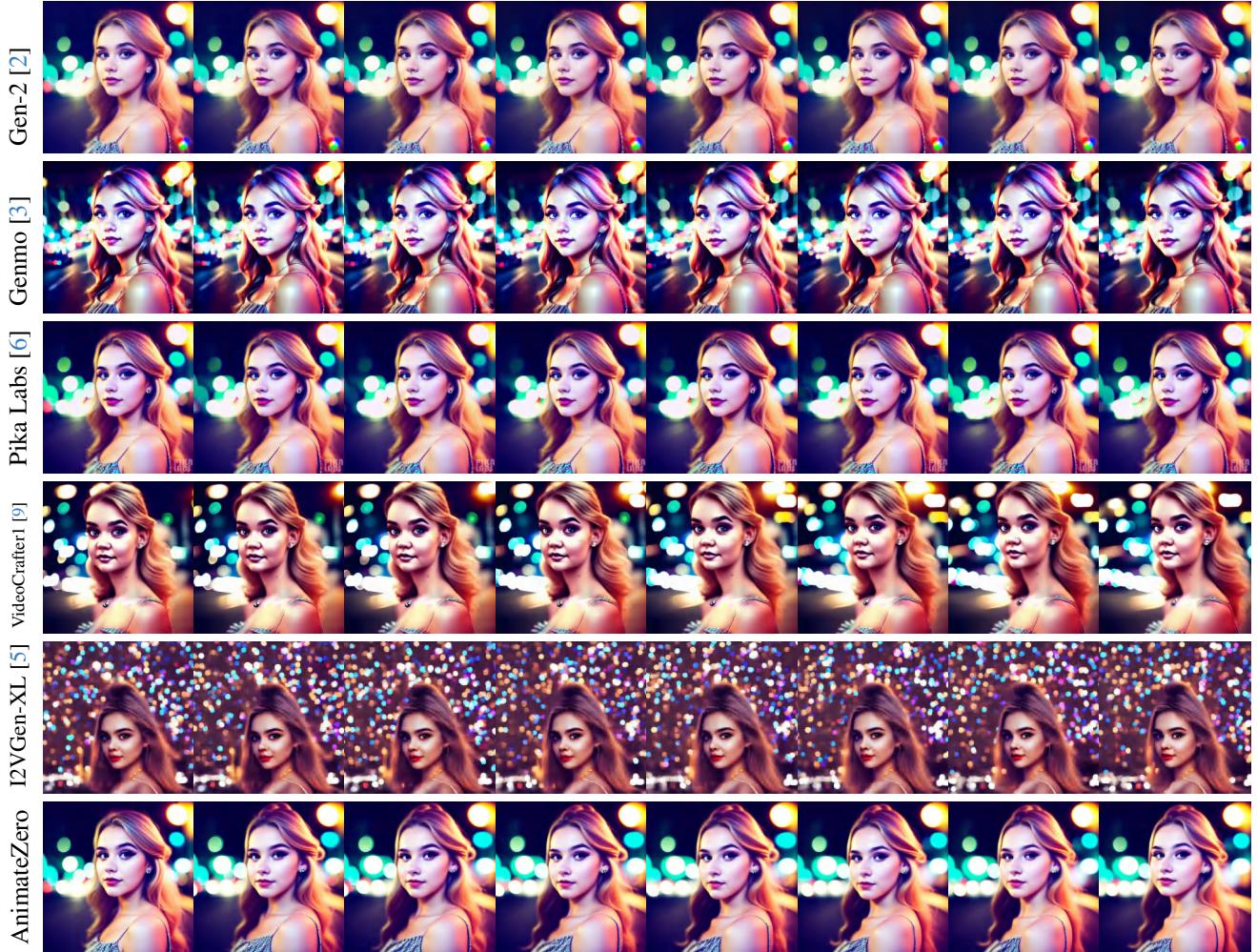


Figure 12. Static frames sequences in Fig. 4.



"Igirl, brown hair; a lot of white flowers, leaf, blurry foreground, ..."

Figure 13. Static frames sequences in Fig. 5 (part 1).



“closeup face photo of 18 y.o swedish woman in dress, makeup, night city street, motion blur, ...”

Figure 14. Static frames sequences in Fig. 5 (part 2).



Figure 15. Static frames sequences in Fig. 7.

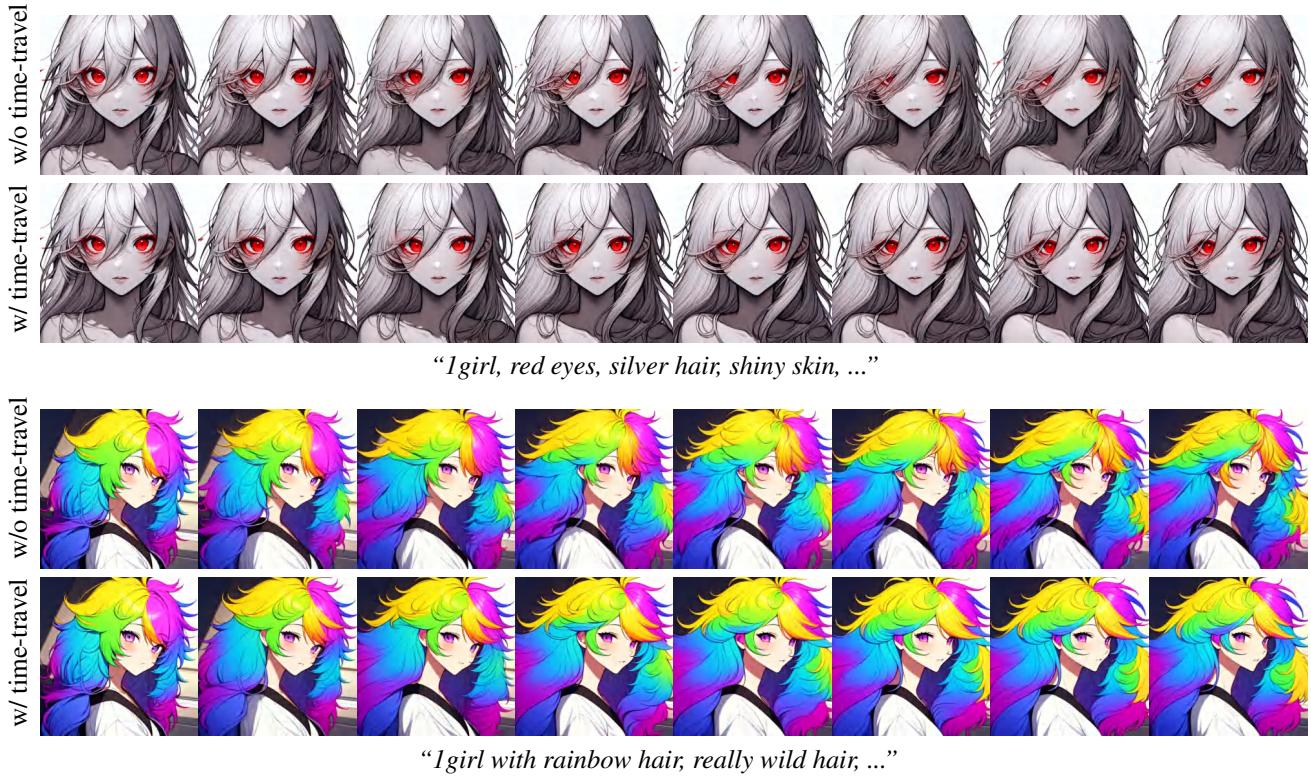


Figure 16. Static frames sequences in Fig. 8.