ANIMATEDIFF: ANIMATE YOUR PERSONALIZED TEXT-TO-IMAGE DIFFUSION MODELS WITHOUT SPECIFIC TUNING

Yuwei Guo[1] **Ceyuan Yang**[2†] **Anyi Rao**[3] **Zhengyang Liang**[2] **Yaohui Wang**[2]
Yu Qiao[2] **Maneesh Agrawala**[3] **Dahua Lin**[1,2] **Bo Dai**[2]
[1]The Chinese University of Hong Kong [2]Shanghai Artificial Intelligence Laboratory
[3]Stanford University

(cartoon) *1boy, dark skin, playing guitar, concert, . . .* ***(oil painting)*** *black pearl pirate ship, night time, sea, . . .* ***(realistic)*** *a Lamborghini on road, fireworks, high detail, . . .*

zoom-in ***rolling*** ***zoom-out + rolling*** ***right + up***

Figure 1: AnimateDiff directly turns existing personalized text-to-image (T2I) models to the corresponding animation generators with a pre-trained motion module. ***First row***: results by combining AnimateDiff with three personalized T2Is in different domains; ***Second row***: results of further combining AnimateDiff with MotionLoRA (s) to achieve shot type controls. *Best viewed with Acrobat Reader. Click the images to play the animation clips.*

ABSTRACT

With the advance of text-to-image (T2I) diffusion models (*e.g.*, Stable Diffusion) and corresponding personalization techniques such as DreamBooth and LoRA, everyone can manifest their imagination into high-quality images at an affordable cost. However, adding motion dynamics to existing high-quality personalized T2Is and enabling them to generate animations remains an open challenge. In this paper, we present AnimateDiff, a practical framework for animating personalized T2I models without requiring model-specific tuning. At the core of our framework is a plug-and-play motion module that can be trained once and seamlessly integrated into any personalized T2Is originating from the same base T2I. Through our proposed training strategy, the motion module effectively learns transferable motion priors from real-world videos. Once trained, the motion module can be inserted into a personalized T2I model to form a personalized animation

[†]Corresponding Author.

generator. We further propose MotionLoRA, a lightweight fine-tuning technique for AnimateDiff that enables a pre-trained motion module to adapt to new motion patterns, such as different shot types, at a low training and data collection cost. We evaluate AnimateDiff and MotionLoRA on several public representative personalized T2I models collected from the community. The results demonstrate that our approaches help these models generate temporally smooth animation clips while preserving the visual quality and motion diversity. Codes and pre-trained weights are available at https://github.com/guoyww/AnimateDiff.

1 INTRODUCTION

Text-to-image (T2I) diffusion models (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022) have greatly empowered artists and amateurs to create visual content using text prompts. To further stimulate the creativity of existing T2I models, lightweight personalization methods, such as DreamBooth (Ruiz et al., 2023) and LoRA (Hu et al., 2021) have been proposed. These methods enable customized fine-tuning on small datasets using consumer-grade hardware such as a laptop with an RTX3080, thereby allowing users to adapt a base T2I model to new domains and improve visual quality at a relatively low cost. Consequently, a large community of AI artists and amateurs has contributed numerous personalized models on model-sharing platforms such as Civitai (2022) and Hugging Face (2022). While these personalized T2I models can generate remarkable visual quality, their outputs are limited to static images. On the other hand, the ability to generate animations is more desirable in real-world production, such as in the movie and cartoon industries. In this work, we aim to directly transform existing high-quality personalized T2I models into animation generators without requiring model-specific fine-tuning, which is often impractical in terms of computation and data collection costs for amateur users.

We present AnimateDiff, an effective pipeline for addressing the problem of animating personalized T2Is while preserving their visual quality and domain knowledge. The core of AnimateDiff is an approach for training a plug-and-play motion module that learns reasonable motion priors from video datasets, such as WebVid-10M (Bain et al., 2021). At inference time, the trained motion module can be directly integrated into personalized T2Is and produce smooth and visually appealing animations without requiring specific tuning. The training of the motion module in AnimateDiff consists of three stages. Firstly, we fine-tune a domain adapter on the base T2I to align with the visual distribution of the target video dataset. This preliminary step guarantees the motion module concentrates on learning the motion priors rather than pixel-level details from the training videos. Secondly, we inflate the base T2I together with the domain adapter and introduce a newly initialized motion module for motion modeling. We then optimize this module on videos while keeping the domain adapter and base T2I weights fixed. By doing so, the motion module learns generalized motion priors and can, via module insertion, enable other personalized T2Is to generate smooth and appealing animations aligned with their personalized domains. The third stage of AnimateDiff, also dubbed as MotionLoRA, aims to adapt the pre-trained motion module to specific motion patterns with a small number of reference videos and training iterations. We achieve this by fine-tuning the motion module with the aid of Low-Rank Adaptation (LoRA) (Hu et al., 2021). Remarkably, adapting to a new motion pattern can be achieved with as few as 50 reference videos. Moreover, a MotionLoRA model requires only approximately 30M of additional storage space, further enhancing the efficiency of model sharing. This efficiency is particularly valuable for users who are unable to bear the expensive costs of pre-training but desire to fine-tune the motion module for specific effects.

We evaluate the performance of AnimateDiff and MotionLoRA on a diverse set of personalized T2I models collected from model-sharing platforms (Civitai, 2022; Hugging Face, 2022). These models encompass a wide spectrum of domains, ranging from 2D cartoons to realistic photographs, thereby forming a comprehensive benchmark for our evaluation. The results of our experiments demonstrate promising outcomes. In practice, we also found that a Transformer (Vaswani et al., 2017) architecture along the temporal axis is adequate for capturing appropriate motion priors. We also demonstrate that our motion module can be seamlessly integrated with existing content-controlling approaches (Zhang et al., 2023; Mou et al., 2023) such as ControlNet without requiring additional training, enabling AnimateDiff for controllable animation generation.

In summary, (1) we present AnimateDiff, a practical pipeline that enables the animation generation ability of any personalized T2Is without specific fine-tuning; (2) we verify that a Transformer architecture is adequate for modeling motion priors, which provides valuable insights for video generation; (3) we propose MotionLoRA, a lightweight fine-tuning technique to adapt pre-trained motion modules to new motion patterns; (4) we comprehensively evaluate our approach with representative community models and compare it with both academic baselines and commercial tools such as Gen-2 (2023) and Pika Labs (2023). Furthermore, we showcase its compatibility with existing works for controllable generation.

2 Related Work

Text-to-image diffusion models. Diffusion models (Ho et al., 2020; Dhariwal & Nichol, 2021; Song et al., 2020) for text-to-image (T2I) generation (Gu et al., 2022; Mokady et al., 2023; Podell et al., 2023; Ding et al., 2021; Zhou et al., 2022b; Ramesh et al., 2021; Li et al., 2022) have gained significant attention in both academic and non-academic communities recently. GLIDE (Nichol et al., 2021) introduced text conditions and demonstrated that incorporating classifier guidance leads to more pleasing results. DALL-E2 (Ramesh et al., 2022) improves text-image alignment by leveraging the CLIP (Radford et al., 2021) joint feature space. Imagen (Saharia et al., 2022) incorporates a large language model (Raffel et al., 2020) and a cascade architecture to achieve photorealistic results. Latent Diffusion Model (Rombach et al., 2022), also known as Stable Diffusion, moves the diffusion process to the latent space of an auto-encoder to enhance efficiency. eDiff-I (Balaji et al., 2022) employs an ensemble of diffusion models specialized for different generation stages.

Personalizing T2I models. To facilitate the creation with pre-trained T2Is, many works focus on efficient model personalization (Shi et al., 2023; Lu et al., 2023; Dong et al., 2022; Kumari et al., 2023), *i.e.*, introducing concepts or styles to the base T2I using reference images. The most straightforward approach to achieve this is complete fine-tuning of the model. Despite its potential to significantly enhance overall quality, this practice can lead to catastrophic forgetting (Kirkpatrick et al., 2017; French, 1999) when the reference image set is small. Instead, DreamBooth (Ruiz et al., 2023) fine-tunes the entire network with preservation loss and uses only a few images. Textual Inversion (Gal et al., 2022) optimize a token embedding for each new concept. Low-Rank Adaptation (LoRA) (Hu et al., 2021) facilitates the above fine-tuning process by introducing additional LoRA layers to the base T2I and optimizing only the weight residuals. There are also encoder-based approaches that address the personalization problem (Gal et al., 2023; Jia et al., 2023). In our work, we focus on tuning-based methods, including overall fine-tuning, DreamBooth (Ruiz et al., 2023), and LoRA (Hu et al., 2021), as they preserve the original feature space of the base T2I.

Animating personalized T2Is. There are not many existing works regarding animating personalized T2Is. Text2Cinemagraph (Mahapatra et al., 2023) proposed to generate cinematography via flow prediction. In the field of video generation, it is common to extend a pre-trained T2I with temporal structures. Existing works (Esser et al., 2023; Zhou et al., 2022a; Singer et al., 2022; Ho et al., 2022b,a; Ruan et al., 2023; Luo et al., 2023; Yin et al., 2023b,a; Wang et al., 2023b; Hong et al., 2022; Luo et al., 2023) mostly update all parameters and modify the feature space of the original T2I and is not compatible with personalized ones. Align-Your-Latents (Blattmann et al., 2023) shows that the frozen image layers in a general video generator can be personalized. Recently, some video generation approaches have shown promising results in animating a personalized T2I model. Tune-a-Video (Wu et al., 2023) fine-tune a small number of parameters on a single video. Text2Video-Zero (Khachatryan et al., 2023) introduces a training-free method to animate a pre-trained T2I via latent wrapping based on a pre-defined affine matrix.

3 Preliminary

We introduce the preliminary of Stable Diffusion (Rombach et al., 2022), the base T2I model used in our work, and Low-Rank Adaptation (LoRA) (Hu et al., 2021), which helps understand the domain adapter (Sec. 4.1) and MotionLoRA (Sec. 4.3) in AnimateDiff.

Stable Diffusion. We chose Stable Diffusion (SD) as the base T2I model in this paper since it is open-sourced and has a well-developed community with many high-quality personalized T2I models for evaluation. SD performs the diffusion process within the latent space of a pre-trained autoen-

coder $\mathcal{E}(\cdot)$ and $\mathcal{D}(\cdot)$. In training, an encoded image $z_0 = \mathcal{E}(x_0)$ is perturbed to z_t by the forword diffusion:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \ \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

for $t = 1, \ldots, T$, where pre-defined $\bar{\alpha}_t$ determines the noise strength at step t. The denoising network $\epsilon_\theta(\cdot)$ learns to reverse this process by predicting the added noise, encouraged by an MSE loss:

$$\mathcal{L} = \mathbb{E}_{\mathcal{E}(x_0), y, \epsilon \sim \mathcal{N}(0,I), t} \left[\| \epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y)) \|_2^2 \right], \tag{2}$$

where y is the text prompt corresponding to x_0; $\tau_\theta(\cdot)$ is a text encoder mapping the prompt to a vector sequence. In SD, $\epsilon_\theta(\cdot)$ is implemented as a UNet (Ronneberger et al., 2015) consisting of pairs of down/up sample blocks at four resolution levels, as well as a middle block. Each network block consists of ResNet (He et al., 2016), spatial self-attention layers, and cross-attention layers that introduce text conditions.

Low-rank adaptation (LoRA). LoRA (Hu et al., 2021) is an approach that accelerates the fine-tuning of large models and is first proposed for language model adaptation. Instead of retraining all model parameters, LoRA adds pairs of rank-decomposition matrices and optimizes only these newly introduced weights. By limiting the trainable parameters and keeping the original weights frozen, LoRA is less likely to cause catastrophic forgetting (Kirkpatrick et al., 2017). Concretely, the rank-decomposition matrices serve as the residual of the pre-trained model weights $\mathcal{W} \in \mathbb{R}^{m \times n}$. The new model weight with LoRA is

$$\mathcal{W}' = \mathcal{W} + \Delta\mathcal{W} = \mathcal{W} + AB^T, \tag{3}$$

where $A \in \mathbb{R}^{m \times r}$, $B \in \mathbb{R}^{n \times r}$ are a pair of rank-decomposition matrices, r is a hyper-parameter, which is referred to as the rank of LoRA layers. In practice, LoRA is only applied to attention layers, further reducing the cost and storage for model fine-tuning.

4 AnimateDiff

The core of our method is learning transferable motion priors from video data, which can be applied to personalized T2Is without specific tuning. As shown in Fig. 2, at inference time, our motion module (blue) and the *optional* MotionLoRA (green) can be directly inserted into a personalized T2I to constitute the animation generator, which subsequently generates animations via an iterative denoising process.

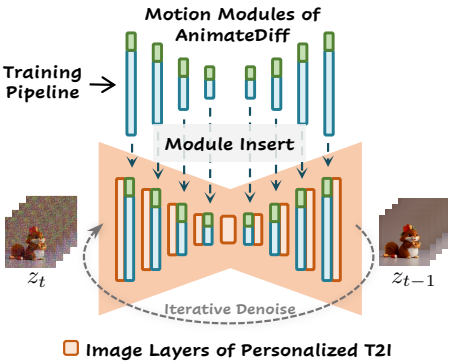


Figure 2: Inference pipeline.

We achieve this by training three components of AnimateDiff, namely domain adapter, motion module, and MotionLoRA. The domain adapter in Sec. 4.1 is only used in the training to alleviate the negative effects caused by the visual distribution gap between the base T2I pre-training data and our video training data; the motion module in Sec. 4.2 is for learning the motion priors; and the MotionLoRA in Sec. 4.3, which is *optional* in the case of general animation, is for adapting pre-trained motion modules to new motion patterns. Sec.4.4 elaborates on the training (Fig. 3) and inference of AnimateDiff.

4.1 Alleviate Negative Effects from Training Data with Domain Adapter

Due to the difficulty in collection, the visual quality of publicly available video training datasets is much lower than their image counterparts. For example, the contents of the video dataset WebVid (Bain et al., 2021) are mostly real-world recordings, whereas the image dataset LAION-Aesthetic (Schuhmann et al., 2022) contains higher-quality contents, including artistic paintings and professional photography. Moreover, when treated individually as images, each video frame can contain motion blur, compression artifacts, and watermarks. Therefore, there is a non-negligible quality domain gap between the high-quality image dataset used to train the base T2I and the target

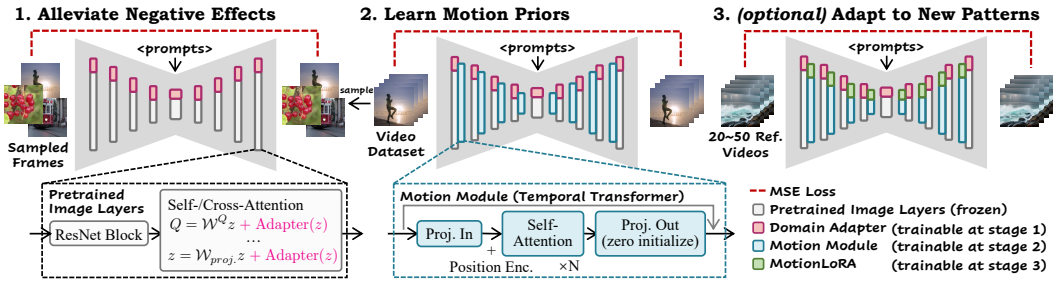


Figure 3: **Training pipeline of AnimateDiff.** AnimateDiff consists of three training stages for the corresponding component modules. Firstly, a domain adapter (Sec. 4.1) is trained to alleviate the negative effects caused by training videos. Secondly, a motion module (Sec. 4.2) is inserted and trained on videos to learn general motion priors. Lastly, MotionLoRA (Sec. 4.3) is trained on a few reference videos to adapt the pre-trained motion module to new motion patterns.

video dataset we use for learning the motion priors. We argue that such a gap can limit the quality of the animation generation pipeline when trained directly on the raw video data.

To avoid learning this quality discrepancy as part of our motion module and preserve the knowledge of the base T2I, we propose to **fit the domain information to a separate network**, dubbed as domain adapter. We drop the domain adapter at inference time and show that this practice helps reduce the negative effects caused by the domain gap mentioned above. We implement the domain adapter layers with LoRA (Hu et al., 2021) and insert them into the self-/cross-attention layers in the base T2I, as shown in Fig. 3. Take query (Q) projection as an example. The internal feature z after projection becomes

$$Q = \mathcal{W}^Q z + \text{AdapterLayer}(z) = \mathcal{W}^Q z + \alpha \cdot AB^T z, \tag{4}$$

where $\alpha = 1$ is a scalar and can be adjusted to other values at inference time (set to 0 to remove the effects of domain adapter totally). We then optimize only the parameters of the domain adapter on static frames randomly sampled from video datasets with the same objective in Eq. (2).

4.2 Learn Motion Priors with Motion Module

To model motion dynamics along the temporal dimension on top of a pre-trained T2I, we must 1) inflate the 2-dimensional diffusion model to deal with 3-dimensional video data and 2) design a sub-module to enable efficient information exchange along the temporal axis.

Network Inflation. The pre-trained image layers in the base T2I model capture high-quality content priors. To utilize the knowledge, a preferable way for network inflation is to let these image layers independently deal with video frames. To achieve this, we adopt a practice similar to recent works (Ho et al., 2022b; Wu et al., 2023; Blattmann et al., 2023), and modify the model so that it takes 5D video tensors $x \in \mathbb{R}^{b \times c \times f \times h \times w}$ as input, where b and f represent batch axis and frame-time axis respectively. When the internal feature maps go through image layers, the temporal axis f is ignored by being reshaped into the b axis, allowing the network to process each frame independently. We then reshape the feature map to the 5D tensor after the image layer. On the other hand, our newly inserted motion module ignores the spatial axis by reshaping h, w into b and then reshaping back after the module.

Module Design. Recent works on video generation have explored many designs for temporal modeling. In AnimateDiff, we adopt the Transformer (Vaswani et al., 2017) architecture as our motion module design, and make minor modifications to adapt it to operate along the temporal axis, which we refer to as "temporal Transformer" in the following sections. We experimentally found this design is adequate for modeling motion priors. As illustrated in Fig. 3, the temporal Transformer consists of several self-attention blocks along the temporal axis, with sinusoidal position encoding to encode the location of each frame in the animation. As mentioned above, the input of the motion module is the reshaped feature map whose spatial dimensions are merged into the batch axis. When we divide the reshaped feature map along the temporal axis, it can be regarded as vector sequences with length of f, *i.e.*, $\{z_1, ..., z_f; z_i \in \mathbb{R}^{(b \times h \times w) \times c}\}$. The vectors will then be projected and go

through several self-attention blocks, *i.e.*

$$z_{out} = \text{Attention}(Q, K, V) = \text{Softmax}(QK^T/\sqrt{c}) \cdot V, \tag{5}$$

where $Q = W^Q z$, $K = W^K z$, and $V = W^V z$ are three separated projections. The attention mechanism enables the generation of the current frame to incorporate information from other frames. As a result, instead of generating each frame individually, the T2I model inflated with our motion module learns to capture the changes of visual content over time, which constitute the motion dynamics in an animation clip. Note that sinusoidal position encoding added before the self-attention is essential; otherwise, the module is not aware of the frame order in the animation. To avoid any harmful effects that the additional module might introduce, we zero initialize (Zhang et al., 2023) the output projection layers of the temporal Transformer and add a residual connection so that the motion module is an identity mapping at the beginning of training.

4.3 Adapt to New Motion Patterns with MotionLoRA

While the pre-trained motion module captures general motion priors, a question arises when we need to effectively adapt it to new motion patterns such as camera zooming, panning and rolling, *etc.*, with a small number of reference videos and training iterations. Such efficiency is essential for users who cannot afford expensive pre-training costs but would like to fine-tune the motion module for specific effects. Here comes the last stage of AnimateDiff, also dubbed as MotionLoRA (Fig. 3), an efficient fine-tuning approach for **motion personalization**. Considering the architecture of the motion module and the limited number of reference videos, we add LoRA layers to the self-attention layers of the motion module in the inflated model described in Sec. 4.2, then train these LoRA layers on the reference videos of new motion patterns.

We experiment with several shot types and get the reference videos via rule-based data augmentation. For instance, to get videos with zooming effects, we augment the videos by gradually reducing (zoom-in) or enlarging (zoom-out) the cropping area of video frames along the temporal axis. We demonstrate that our MotionLoRA can achieve promising results even with as few as $20 \sim 50$ reference videos, 2,000 training iterations (around $1 \sim 2$ hours) as well as about 30M storage space, enabling efficient model tuning and sharing among users. Benefited by the low-rank property, MotionLoRA also has the composition capability. Namely, individually trained MotionLoRA models can be combined to achieve composed motion effects at inference time.

4.4 AnimateDiff in Practice

Training. As illustrated in Fig. 3, AnimateDiff consists of three trainable component modules to learn transferable motion priors. Their training objectives are slightly different. The domain adapter is trained with the original objective as in Eq. (2). The motion module and MotionLoRA, as part of an animation generator, use a similar objective with minor modifications to accommodate higher dimension video data. Concretely, a video data batch $x_0^{1:f} \in \mathbb{R}^{b \times c \times f \times h \times w}$ is first encoded into the latent codes $z_0^{1:f}$ frame-wisely via the pre-trained auto-encoder of SD. The latent codes are then noised using the defined forward diffusion schedule as in Eq. (1)

$$z_t^{1:f} = \sqrt{\bar{\alpha}_t} z_0^{1:f} + \sqrt{1 - \bar{\alpha}_t} \epsilon^{1:f}. \tag{6}$$

The inflated model inputs the noised latent codes and corresponding text prompts and predicts the added noises. The final training objective of our motion modeling module is:

$$\mathcal{L} = \mathbb{E}_{\mathcal{E}(x_0^{1:f}), y, \epsilon^{1:f} \sim \mathcal{N}(0,I), t} \left[\| \epsilon - \epsilon_\theta(z_t^{1:f}, t, \tau_\theta(y)) \|_2^2 \right]. \tag{7}$$

It's worth noting that when training the domain adapter, the motion module, and the MotionLoRA, parameters outside the trainable part remain frozen.

Inference. At inference time (Fig. 2), the personalized T2I model will first be inflated in the same way discussed in Section 4.2, then injected with the motion module for general animation generation, and the *optional* MotionLoRA for generating animation with personalized motion. As for the domain adapter, instead of simply dropping it during the inference time, in practice, we can also inject it into the personalized T2I model and adjust its contribution by changing the scaler α in Eq. (4). An ablation study on the value of α is conducted in experiments. Finally, the animation frames can be obtained by performing the reverse diffusion process and decoding the latent codes.

Figure 4: **Qualitative Result.** Each sample corresponds to a distinct personalized T2I. *Best viewed with Acrobat Reader. Click the images to play the animation clips.*

5 Experiments

We implement AnimateDiff upon Stable Diffusion V1.5 and train motion module using the WebVid-10M (Bain et al., 2021) dataset. Detailed configurations can be found in supplementary materials.

5.1 Qualitative Results

Evaluate on community models. We evaluated the AnimateDiff with a diverse set of representative personalized T2Is collected from Civitai (2022). These personalized T2Is encompass a wide range of domains, thus serving as a comprehensive benchmark. Since personalized domains in these T2Is only respond to certain "trigger words", we abstain from using common text prompts but refer to the model homepage to construct the evaluation prompts. In Fig. 4, we show eight qualitative results of AnimateDiff. Each sample corresponds to a distinct personalized T2I. In the second row of Figure 1, we present the outcomes obtained by integrating AnimateDiff with MotionLoRA to achieve shot type controls. The last two samples exhibit the composition capability of MotionLoRA, achieved by linearly combining the individually trained weights.

Compare with baselines. In the absence of existing methods specifically designed for animating personalized T2Is, we compare our method with two recent works in video generation that can be adapted for this task: **1) Text2Video-Zero** (Khachatryan et al., 2023) and **2) Tune-a-Video** (Wu et al., 2023). We also compare AnimateDiff with two commercial tools: **3) Gen-2 (2023)** for text-to-video generation, and **4) Pika Labs (2023)** for image animation. The results are shown in Fig. 5.

5.2 Quantitative Comparison

We conduct the quantitative comparison through user study and CLIP metrics. The comparison focuses on three key aspects: *text alignment*, *domain similarity*, and *motion smoothness*. The results are shown in Table 1. Detailed implementations can be found in supplementary materials.

User study. In the user study, we generate animations using all three methods based on the same personalized T2I models. Participants are then asked to individually rank the results based on the

Figure 5: **Qualitative Comparison.** *Best viewed with Acrobat Reader. Click the images to play the animation clips.*

Table 1: Quantitative comparison. A higher score indicates superior performance.

Method	User Study (↑)			CLIP Metric (↑)		
	Text.	Domain.	Smooth.	Text.	Domain.	Smooth.
Text2Video-Zero	1.620	**2.620**	1.560	32.04	84.84	96.57
Tune-a-Video	2.180	1.100	1.615	**35.98**	80.68	97.42
Ours	**2.210**	2.280	**2.825**	31.39	**87.29**	**98.00**

above three aspects. We use the Average User Ranking (AUR) as a preference metric where a higher score indicates superior performance. Note that the corresponding prompts and images are provided for reference for text alignment and domain similarity evaluation.

CLIP metric. We also employed the CLIP (Radford et al., 2021) metric, following the approach taken by previous studies (Wu et al., 2023; Khachatryan et al., 2023). When evaluating domain similarity, it is important to note that the CLIP score was computed between the animation frames and the reference images generated using the personalized T2Is.

5.3 Ablative Study

Domain adapter. To investigate the impact of the domain adapter in AnimateDiff, we conducted a study by adjusting the scaler in the adapter layers during inference, ranging from 1 (full impact) to 0 (complete removal). As illustrated in Figure 6, as the scaler of the adapter decreases, there is an improvement in overall visual quality, accompanied by a reduction in the visual content distribution learned from the video dataset (the watermark in the case of WebVid (Bain et al., 2021)). These results indicate the successful role of the domain adapter in enhancing the visual quality of AnimateDiff by alleviating the motion module from learning the visual distribution gap.

Motion module design. We compare our motion module design of the temporal Transformer with its full convolution counterpart, which is motivated by the fact that both designs are widely employed in recent works on video generation. We replace the temporal attention with 1D temporal convolution and ensured that the two model parameters were closely aligned. As depicted in supplementary materials, the convolutional motion module aligns all frames to be identical but does not incorporate any motion compared to the Transformer architecture.

Efficiency of MotionLoRA. The efficiency of MotionLoRA in AnimateDiff was examined in terms of *parameter efficiency* and *data efficiency*. Parameter efficiency is crucial for efficient model training and sharing among users, while data efficiency is essential for real-world applications where collecting an adequate number of reference videos for specific motion patterns may be challenging.

To investigate these aspects, we trained multiple MotionLoRA models with varying parameter scales and reference video quantities. In Fig. 7, the first two samples demonstrate that MotionLoRA is capable of learning new camera motions (*e.g.*, zoom-in) with a small parameter scale while maintaining comparable motion quality. Furthermore, even with a modest number of reference videos (*e.g.*, $N = 50$), the model successfully learns the desired motion patterns. However, when the number of reference videos is excessively limited (*e.g.*, $N = 5$), significant degradation in quality is observed,

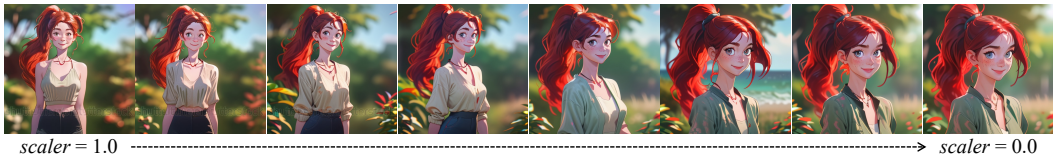


Figure 6: **Ablation on domain adapter.** We adjust the scaler of the adapter from 1 to 0 to gradually remove its effects. In this figure, we show the first frame of the generated animation.

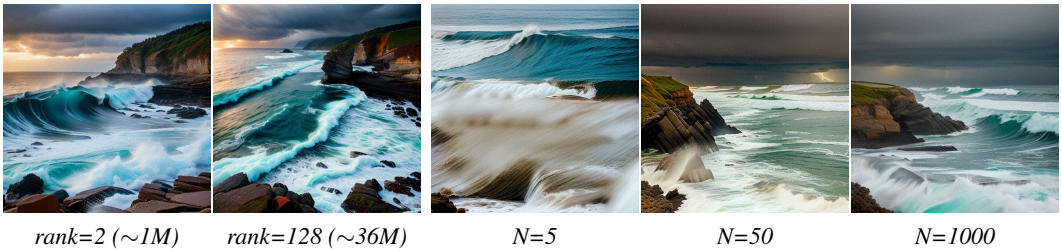
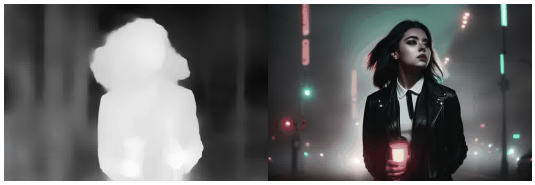


Figure 7: **Ablation on MotionLoRA's efficiency.** *Two samples on the left*: with different network rank; *Three samples on the right*: with different numbers of reference videos. *Best viewed with Acrobat Reader. Click the images to play the animation clips.*

suggesting that MotionLoRA encounters difficulties in learning shared motion patterns and instead relies on capturing texture information from the reference videos.

5.4 Controllable generation.

The separated learning of visual content and motion priors in AnimateDiff enables the direct application of existing content control approaches for controllable generation. To demonstrate this capability, we combined AnimateDiff with ControlNet (Zhang et al., 2023) to control the generation with extracted depth map sequence. In contrast to recent video editing techniques (Ceylan et al., 2023; Wang et al., 2023a) that employ DDIM (Song et al., 2020) inversion to obtain smoothed latent sequences, we generate animations from randomly sampled noise. As illustrated in Figure 8, our results exhibit meticulous motion details (such as hair and facial expressions) and high visual quality.



city street, neon, fog, closeup portrait photo of young woman in dark clothes, . . .

Figure 8: **Controllable generation.** *Best viewed with Acrobat Reader. Click the images to play the animation clips.*

6 Conclusion

In this paper, we present AnimateDiff, a practical pipeline directly turning personalized text-to-image (T2I) models for animation generation once and for all, without compromising quality or losing pre-learned domain knowledge. To accomplish this, we design three component modules in AnimateDiff to learn meaningful motion priors while alleviating visual quality degradation and enabling motion personalization with a lightweight fine-tuning technique named MotionLoRA. Once trained, our motion module can be integrated into other personalized T2Is to generate animated images with natural and coherent motions while remaining faithful to the personalized domain. Extensive evaluation with various personalized T2I models also validates the effectiveness and generalizability of our AnimateDiff and MotionLoRA. Furthermore, we demonstrate the compatibility of our method with existing content-controlling approaches, enabling controllable generation without incurring additional training costs. Overall, AnimateDiff provides an effective baseline for personalized animation and holds significant potential for a wide range of applications.

7 ETHICS STATEMENT

We strongly condemn the misuse of generative AI to create content that harms individuals or spreads misinformation. However, we acknowledge the potential for our method to be misused since it primarily focuses on animation and can generate human-related content. It is also important to highlight that our method incorporates personalized text-to-image models developed by other artists. These models may contain inappropriate content and can be used with our method.

To address these concerns, we uphold the highest ethical standards in our research, including adhering to legal frameworks, respecting privacy rights, and encouraging the generation of positive content. Furthermore, we believe that introducing an additional content safety checker, similar to that in Stable Diffusion (Rombach et al., 2022), could potentially resolve this issue.

8 REPRODUCIBILITY STATEMENT

We provide comprehensive implementation details for the training and inference of our method in supplementary materials, aiming to enhance the reproducibility of our approach. We also make both the code and pre-trained weights open-sourced to facilitate further investigation and exploration.

ACKNOWLEDGEMENT

This project is funded in part by Shanghai AI Laboratory (P23KS00020, 2022ZD0160201), CUHK Interdisciplinary AI Research Institute, and the Centre for Perceptual and Interactive Intelligence (CPIl) Ltd under the Innovation and Technology Commission (ITC)'s InnoHK.

REFERENCES

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 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*, pp. 1728–1738, 2021.

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

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

Duygu Ceylan, Chun-Hao Paul Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. *arXiv preprint arXiv:2303.12688*, 2023.

Civitai. Civitai. https://civitai.com/, 2022.

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

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

Ziyi Dong, Pengxu Wei, and Liang Lin. Dreamartist: Towards controllable one-shot text-to-image generation via contrastive prompt-tuning. *arXiv preprint arXiv:2211.11337*, 2022.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. *arXiv preprint arXiv:2302.03011*, 2023.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Rinon Gal, Moab Arar, Yuval Atzmon, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Designing an encoder for fast personalization of text-to-image models. *arXiv preprint arXiv:2302.12228*, 2023.

Gen-2. Gen-2: The next step forward for generative ai. https://research.runwayml.com/gen2/, 2023.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

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*, 2022a.

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

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hugging Face. Huggingface. https://huggingface.co/, 2022.

Xuhui Jia, Yang Zhao, Kelvin CK Chan, Yandong Li, Han Zhang, Boqing Gong, Tingbo Hou, Huisheng Wang, and Yu-Chuan Su. Taming encoder for zero fine-tuning image customization with text-to-image diffusion models. *arXiv preprint arXiv:2304.02642*, 2023.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *IEEE International Conference on Computer Vision (ICCV)*, 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1931–1941, 2023.

Wei Li, Xue Xu, Xinyan Xiao, Jiachen Liu, Hu Yang, Guohao Li, Zhanpeng Wang, Zhifan Feng, Qiaoqiao She, Yajuan Lyu, et al. Upainting: Unified text-to-image diffusion generation with cross-modal guidance. *arXiv preprint arXiv:2210.16031*, 2022.

Haoming Lu, Hazarapet Tunanyan, Kai Wang, Shant Navasardyan, Zhangyang Wang, and Humphrey Shi. Specialist diffusion: Plug-and-play sample-efficient fine-tuning of text-to-image diffusion models to learn any unseen style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14267–14276, 2023.

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*, pp. 10209–10218, 2023.

Aniruddha Mahapatra, Aliaksandr Siarohin, Hsin-Ying Lee, Sergey Tulyakov, and Jun-Yan Zhu. Text-guided synthesis of eulerian cinemagraphs, 2023.

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*, pp. 6038–6047, 2023.

Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Pika Labs. Pika labs. https://www.pika.art/, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

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*, pp. 8748–8763. PMLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

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

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*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10219–10228, 2023.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

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. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. *arXiv preprint arXiv:2304.03411*, 2023.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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

Wen Wang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023a.

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, Yuwei Guo, Tianxing Wu, Chenyang Si, Yuming Jiang, Cunjian Chen, Chen Change Loy, Bo Dai, Dahua Lin, Yu Qiao, and Ziwei Liu. Lavie: High-quality video generation with cascaded latent diffusion models, 2023b.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *IEEE International Conference on Computer Vision (ICCV)*, 2023.

Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023a.

Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346*, 2023b.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *IEEE International Conference on Computer Vision (ICCV)*, 2023.

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*, 2022a.

Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Towards language-free training for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17907–17917, 2022b.