MagicAnimate: Temporally Consistent Human Image Animation using Diffusion Model

Supplementary Material

Overview

In this supplementary material, we provide additional details and video results for the main paper:

- More descriptions of our method (Sec. 1 and 2) and experiments (Sec. 3, 4, and 5).
- The Project Page showing all the video results (Sec. 6).
- Additional experiments for ablation studies (Sec. 7).
- Discussions of our Limitations (Sec. 8) and Societal Impacts (Sec. 9).

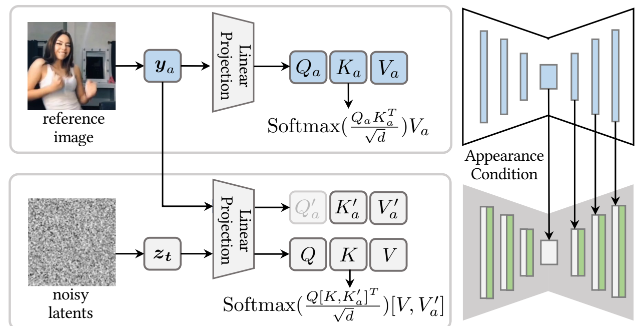

Figure 1. We extract appearance features $\boldsymbol{y}_a$ using our appearance encoder. These appearance features are integrated into the video synthesis backbone through the hybrid self-attention mechanism. In the appearance encoder, $\boldsymbol{y}_a$ is fed into the default self-attention blocks without any modification. To incorporate the appearance condition, we calculate the appearance key K'_a and value V'_a for $\boldsymbol{y}_a$ using the linear projection layers of the video synthesis backbone. Subsequently, we concatenate the keys and values into $[K, K'_a]$ and $[V, V'_a]$ to compute attention scores for video animation synthesis.

1. Appearance Encoder Details

The appearance encoder in our model transforms the reference image I_{ref} into the appearance condition $\boldsymbol{y}_a$. We then integrate this condition information into the video synthesis backbone. In order to preserve the spatial layout of the image and retain information from reference image, we adapt the self-attention mechanism into the hybrid one by querying features from both $\boldsymbol{z_t}$ and $\boldsymbol{y}_a$. The appearance condition method is illustrated in Figure 1. At each denoising step $\boldsymbol{t}$, the reference image I_{ref} is initially encoded into latents using the pretrained VAE encoder [5]. Subsequently, we feed these latents into appearance encoder backbone, *i.e.*, a trainable UNet copy, to obtain $\boldsymbol{y}_a$, which represents the normalized outputs of the first layer in each self-attention Transformer block.

For the appearance encoder branch, we directly feed $\boldsymbol{y}_a$ into the original attention layers without any modification. The Attention(Q_a, K_a, V_a) for $\boldsymbol{y}_a$ is calculated by the default forward pass. As for the integration of appearance condition, we pass $\boldsymbol{y}_a$ to the linear projection layers in the video synthesis backbone to compute K'_a and V'_a. Simultaneously, the noisy latents $\boldsymbol{z_t}$ are also projected into Q, K and V. We concatenate the keys and values, denoted as $[K, K'_a]$ and $[V, V'_a]$, to calculate the self-attention scores for video synthesis.

Through this operation, our method can synthesize animations following the provided motion signal and retain the appearance details from the reference image. This robust ability to preserve appearance not only enhances animation fidelity but also contributes to improved temporal consistency in long-term animations.

2. Stage Training Details

Stage I: Appearance encoder and pose control. To save the computation cost, we employ a multi-stage training strategy for MagicAnimate. In the first stage, we temporarily disable the temporal attention layers because they have not been trained or finetuned on our training videos yet. During this stage, we only optimize the appearance encoder and pose ControlNet, facilitating the motion transfer of the reference image. Our DensePose-based ControlNet is pretrained on human images in LAION [6] dataset. In the training process, two frames are sampled from a long video based on a double beta distribution, following established practices in prior works [10, 12]. The first frame acts as the reference image, and the learning objective is to denoise the noisy latent towards the second image, which serves as the target frame. The denoising process is guided by the DensePose of the target image through our pose ControlNet.

For the image joint training in this stage, we directly use the identical reference image and target image from the large-scale image dataset LAION [6]. In this scenario, DensePose is also estimated from the reference image, transforming the learning objective into the reconstruction of the reference image. Despite the absence of explicit modeling of motion transfer in this iteration of reconstruction, our appearance encoder leverages the diversity of the LAION dataset. Consequently, it learns to preserve the details in reference images more effectively, thereby augmenting the final animation fidelity.

Stage II: Temporal attention layers. In the training phase for temporal attention layers, we freeze the appearance encoder and pose ControlNet. The learning objective in this stage is learning the generation of video under the guidance of a reference image and a pose sequence with K frames. During training, a reference image is uniformly sampled from the video, and K consecutive frames with an interval of 4 are sampled to form the target video. Our temporal attention layers are initialized from the pretrained weights released by prior work [3], which is trained on the WebVid [1] dataset.

For this stage, we introduce image-video joint training as well. In each training iteration, there is a probability of reducing the video length K to 1. When the video length is reduced, the learning objective shits to transferring the reference image into the target pose. This training strategy serves two main purposes: (1) The appearance encoder and pose ControlNet remain frozen in this stage. Through this sampling strategy, we can enforce the temporal attention layers to maintain appearance details encoded by the appearance encoder. (2) Given the limited scale of video datasets, we have the flexibility to sample images from LAION for augmenting the training data.

This technique further enhances the single-frame quality of MagicAnimate. Additionally, considering that the TED-talks dataset exhibits dim lighting conditions and significantly differs in appearance distribution from the LAION dataset, we also sample frames from the TED-talks dataset for image joint training.

3. Implementation Details

We implement MagicAnimate using diffusers[1] library which is built on PyTorch. All experiments are conducted on 8 Nvidia V100 GPUs. In the training stage for the appearance encoder and pose ControlNet, we use a batch size of 8 with a learning rate of 1×10^{-5}. For the image-video joint training, τ_0 is set to 0.2. In the training of temporal attentions, a batch size of 8 is used with a learning rate of 1×10^{-4}. For the image-video joint training of this stage, different sampling thresholds τ_1 and τ_2 are used for different datasets. On TikTok dataset [4], we set τ_1 to 0.2 and τ_2 to 0.2 for all the experiments except for applications. We empirically find that using a smaller τ_1 and τ_2 can improve generalization ability. Thus, we set τ_1 and τ_2 to 0 for application experiments. For the TED-talks [7] dataset, we use a τ_1 of 0.2 and τ_2 of 0.36. In MagicAnimate, K is set to 16 and s is set to 4. The generation resolution is set to 512×512. During training, we only apply horizontal flip augmentation for training videos.

[1] https://github.com/huggingface/diffusers

4. Dataset Preprocessing

We process the video datasets using a standard preprocessing pipeline:

- *Download videos*: We download the original TikTok video frames released by Jafarian *et al*. [4] and keep the complete frames without any crop. For TED-talks [7], we follow their official instructions to download original Youtube videos with the highest possible resolutions. We then crop and truncate the videos into multiple clips based on the official tracklets. Different from the original square crops, we crop and resize the clips into a resolution of 1024×512 to keep a larger field of view, which benefits the estimation of DensePose. We extract all the video clips into frames with 25 fps.
- *Horizontal flip augmentation*: MagicAnimate employs DensePose as motion signal, but DensePose definition is asymmetric and cannot be horizontally flipped. Thus, we flip all of the videos in advance for augmentation and double the dataset scale.
- *Estimate DensePose*: We use the official implementation[2] to estimate DensePose for each video frame. Our motion signal is derived from the visualization of the DensePose segmentation map.
- *Estimate background matting masks*: Because certain baseline methods, such as DisCo [8], require a segmentation mask of the human for foreground-background separation, we estimate background matting masks using PaddleSeg library[3].
- *Spatial crop*: The preprocessing steps mentioned above are applied to the original video frames, which typically have a height-width ratio of around $1 : 2$. We then perform a center crop on all video frames and resize them into 512×512.

Additionally, we make use of human images from LAION [6] for pretraining our DensePose-based ControlNet and for the image-video joint training. Consequently, we estimate the DensePose for each human image in the LAION dataset.

5. Details for PSNR Metrics

In our initial submission, we follow DisCo [8] and use their official implementation[4] to compute PSNR metrics. However, the community found that there exists an overflow issue[5] in their implementation. In the final version, we have fixed this numerical overflow and reported the correct PSNR results.

[2] https://github.com/facebookresearch/detectron2
[3] https://github.com/PaddlePaddle/PaddleSeg
[4] https://github.com/Wangt-CN/DisCo
[5] https://github.com/magic-research/magic-animate/issues/146

6. Video Results

To evaluate the performance of MagicAnimate and all the baselines perceptually, we visualize a comprehensive set of complete video results on our Project Page at https://showlab.github.io/magicanimate. The **video** results showcased on our project page include:

- The high-resolution animation results of MagicAnimate on TikTok dancing dataset.
- Qualitative comparisons between MagicAnimate and baselines on both TikTok and TED-talks datasets.
- The qualitative comparisons for cross-identity animation between MagicAnimate and baselines.
- Applications for unseen domain animation, combination with DALL·E3 [2], and multi-person animation.

To ensure deadline integrity, we have compressed our project page along with all video results. These compressed files are included in our supplementary material submission. Reviewers can also uncompress these files and open our project page with the local browser. This provides evidence that no modifications are made after the supplementary deadline.

7. Additional Ablation Studies

Ablations	L1↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	FIDV↓	FVD↓
DWPose	3.48	17.19	0.689	0.259	38.81	17.86	163.89
DW+Dense	3.21	17.95	0.718	0.240	33.62	20.80	133.44
FullCond	3.38	17.96	0.704	0.248	36.64	21.77	168.99
Ours	3.13	18.22	0.714	0.239	32.09	21.75	179.07

Table 1. More ablation studies, we report L1$\times 10^{-4}$.

In this section, we conduct additional experiments for ablation studies.

Driving signals: Table. 1 shows that using keypoints estimated by DWPose [11] produces lower single-frame quality because keypoints are sparse and less stable than DensePose. Furthermore, we combine these two driving signals by addition. It can be observed that although the combined signal (DWPose+DensePose) achieves better video quality, its single-frame quality is not comparable to ours.

Condition layers: Table. 1 shows that the full (down-mid-up) condition has lower single-frame quality than our mid-up condition. We believe the full appearance condition is too strong, which could reduce pose controllability.

8. Limitations

MagicAnimate achieves state-of-the-art human image animation results and demonstrates strong robustness for unseen data. However, there is still room for improvement in several aspects: (1) Although DensePose provides dense guidance for the animation, there exists flickering and occasional failures for the DensePose estimation method [9]. Therefore, enhancing the robustness and accuracy of the DensePose estimator would contribute to the overall performance of our human image animation. (2) DensePose also lacks control signals for facial and finger details. Integrating a multi-ControlNet could fill this gap and potentially enhance the control capabilities for faces and hands. This enhancement may result in more realistic and detailed animations. (3) While diffusion-based methods offer high-quality results, they are generally less efficient than GAN-based methods due to multiple denoising steps. We believe exploring strategies to improve the efficiency of MagicAnimate could largely enhance its applicability.

9. Potential Negative Societal Impacts

The negative societal impact of this work is the potential misuse of our model for malicious purposes, including the generation of misleading content for misinformation, harassment, or fraudulent activities. Moreover, the datasets employed for training our model might inherently contain biases, such as uneven demographic distributions. Consequently, our model may inadvertently perpetuate these biases present in the training data. It is imperative to exercise caution regarding these biases and address fairness considerations when deploying the model.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 2

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, and Yunxin Jiao. Improving image generation with better captions. https://cdn.openai.com/papers/dall-e-3.pdf, 2023. 3

[3] 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*, 2023. 2

[4] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *CVPR*, 2021. 2

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[6] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv*, 2021. 1, 2

[7] Aliaksandr Siarohin, Oliver Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *CVPR*, 2021. 2

[8] Tan Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zhengyuan Yang, Hanwang Zhang, Zicheng Liu, and Lijuan Wang. Disco: Disentangled control for referring human dance generation in real world. *arXiv*, 2023. 2

[9] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3

[10] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Wenqing Zhang, Song Bai, Jiashi Feng, and Mike Zheng Shou. Pv3d: A 3d generative model for portrait video generation. In *ICLR*, 2022. 1

[11] Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4210–4220, 2023. 3

[12] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2022. 1