# EfficientMT: Efficient Temporal Adaptation for Motion Transfer in Text-to-Video Diffusion Models – Supplementary Material –

Anonymous ICCV submission

Paper ID 10596

## 1. More Visualization Results

We have packaged the qualitative video results into a local webpage. We strongly recommend viewing this webpage for a better visual quality assessment of the videos. Please locate the `webpage` folder in the supplementary files and open the `main.html` file in a browser to view the results.

## 2. Implementation Details

### 2.1. Pretrained T2V Model

**AnimateDiff**[3] AnimateDiff is a text-to-video model derived from an extension of text-to-image models. It directly reuses the pre-trained SD (Stable Diffusion) model and inserts a temporal attention layer after the cross-attention layer to model the temporal relationships in videos. During training, the original SD's weights are frozen, and only the weights of the inserted temporal attention layer are trained using video data, thus enabling video generation capabilities. We used the official open-source code[1], employing the stable-diffusion-v1.5 model weights, along with the v3 version of the motion module structure and weights. To enhance the visual quality, we incorporated a third-party image enhancement LoRA module, RealisticVisionV60B1[2]. This module enhances the realistic visual style of the generated videos, with refined contrast and vivid lighting.

**VideoCrafter2**[1] Videocrafter2 adopts the architecture from ModelScope [5], utilizing 1D temporal convolutions and temporal attention to model sequential information. It aims to overcome the limitations of existing open video datasets by leveraging video data of equivalent quality to achieve better video generation models. It first undergoes pre-training on image datasets, followed by composite training using decoupled enhanced video data. We employed the official open-source code[3] and used the 320x512 version of the text-to-video generation model, VideoCrafter2.

[1] https://github.com/guoyww/AnimateDiff
[2] https://civitai.com/models/4201?modelVersionId=245598
[3] https://github.com/AILab-CVC/VideoCrafter

### 2.2. Training Details

For AnimateDiff, we generate a 16-frame video with a resolution of 512x512 and a frame rate of 8, resulting in a video duration of 2 seconds. When extracting reference features, the timestep of denoising process is set to 400, and send *null* as text condition. During training, the learning rate is set to 0.0001, with a batch size of 1. We employ AdamW optimizer with $\beta_1$ = 0.9, $\beta_2$ = 0.999, and a weight decay of 0.01. Training is conducted on a single NVIDIA A100 40GB GPU, with classifier-free guidance text dropout occurring with a probability of 0.1. The model is trained for 100 epochs on the synthetic data. The entire training costs about 15 hours with approximately 32GB VRAM usage during training. For inference, DDIM sampling is employed with 25 sampling steps and a classifier-free guidance strength of 8. The average inference time per sample on a single A100 GPU is 16 seconds with approximately 16GB VRAM usage.

For VideoCrafter2, we generate a 24-frame video with a resolution of 576x320 and a frame rate of 8, resulting in a video duration of 3 seconds. When extracting reference features, the denoising network's time step is set to 400, with *null* as the input text. The training parameters are similar to those used in the AnimateDiff version. The VRAM usage during training is approximately 27GB. For inference, DDIM sampling is applied with 30 sampling steps and a classifier-free guidance strength of 12. We follow [6, 10, 12] to employ the initialization of DDIM inversion noise to enhance the video consistency, due to the limited performance of the base model itself. The average inference time per pass on a single A100 GPU is 21 seconds, with approximately 13GB VRAM usage.

### 2.3. Baseline Implementation

**ControlVideo**[11] We use the official open-source code of ControlVideo[4] and select the depth map of the reference video as the control condition. A 16-frame video is generated with a frame rate of 8, resulting in a 2-second duration. All other experimental settings follow the default configurations provided in the open-source code.

**VideoComposer**[7] We use the official open-source code of VideoComposer[5] and select the motion vector extracted from the reference video as the control condition. A 16-frame video is generated at a resolution of 256x256 with a total duration of 2 seconds. All experimental settings follow the default configurations provided in the open-source code.

**Control-A-Video**[2] We use the official open-source code of Control-V-Video[6] and select the depth map of the reference video as the control condition. A 16-frame video is generated with a frame rate of 8, resulting in a 2-second duration. All experimental settings follow the default configurations provided in the open-source code.

**MotionDirector**[12] We use the official open-source code based on AnimateDiff[7]. The reference video is used as a single sample for fine-tuning, with the video compressed into a single chunk. The model is capable of generating a 16-frame, 2-second video at a resolution of 384x384. Fine-tuning is conducted for 400 iterations, while all other parameters follow the default settings of the code.

**MOFT**[9] We use the official open-source code of MOFT[8] and modify its testing code to adapt it for the motion transfer task. Specifically, we extract MOFT features for each reference video and remove the region mask during optimization to enable global guidance, ensuring that it performs the same task as other methods. All other parameters follow the default settings of the code.

**MotionClone**[4] We use the official open-source code of MotionClone[9], with all parameters following the default settings in the code.

**MotionInversion**[6] We use the official open-source code of MotionInversion[10], setting the fine-tuning iterations to 200 steps. All other parameters follow the default settings in the code.

**DMT**[10] We use the official open-source code of DMT[11], with all parameters following the default settings in the code.

[4] https://github.com/YBYBZhang/ControlVideo
[5] https://github.com/ali-vilab/videocomposer
[6] https://github.com/Weifeng-Chen/control-a-video
[7] https://github.com/ExponentialML/AnimateDiff-MotionDirector
[8] https://github.com/xizaoqu/MOFT
[9] https://github.com/LPengYang/MotionClone
[10] https://github.com/EnVision-Research/MotionInversion
[11] https://github.com/diffusion-motion-transfer/diffusion-motion-transfer

| Scale | Temporal Consistency (↑) | Text Alignment (↑) | Motion Fidelity (↑) |
|---|---|---|---|
| 50 | 0.9165 | 0.2699 | 0.8110 |
| 100 | 0.9227 | 0.2683 | 0.8274 |
| 150 (Ours) | 0.9291 | **0.2712** | **0.8470** |
| 400 | **0.9324** | 0.2688 | 0.8344 |
| full | 0.9237 | 0.2649 | 0.8278 |

Table 1. **Quantitative evaluation on the effect of the training data scale.**

## 3. Additional Experiments

### 3.1. Comparison with trajectory-based method

Trajectory is a simple control signal that guides motion. We qualitatively compare EfficientMT with the current state-of-the-art open-source trajectory motion control method, MotionCtrl [8]. MotionCtrl is a unified framework for camera motion and trajectory motion transfer. It achieves this functionality by inserting an extension plugin into a pre-trained T2V model, enabling fast end-to-end inference. We extract the primary motion trajectory of the subject from the reference video, and use the trajectory as a condition combined with an editing text for transfer using MotionCtrl. As shown in Figure 1, MotionCtrl can follow the editing text to generate a video with corresponding point trajectories; however, the motion patterns in the generated result differ significantly from those in the reference video. This is primarily due to the sparse trajectory serving as a simple control signal, which struggles to provide fine-grained motion guidance and align with the fine-grained motion patterns. In contrast, our method does not require the extraction of additional control signals. Instead, it learns implicit motion patterns from the reference video in an adaptive manner, leading to motion transfer results that closely match the reference video.

### 3.2. Effect of training dataset scale

We further explored the effect of the scale of the training dataset. From the 150 selected training samples, we randomly chose 50 and 100 samples to form small datasets, and we also retrieved 400 lower-quality samples from the constructed data to form an extended dataset. We then used these datasets to train EfficientMT. The quantitative evaluation results from training with different data scales are shown in Table 1. We found that as the amount of data increased in the early stage, the model's quantitative metrics gradually improved. The model performance peaked when the data size reached 150. However, when using the lower-quality extended dataset, the model metrics fluctuated and were generally worse than those with 150 samples. This is primarily due to the interference caused by low-quality samples. Additionally, training with the full dataset led to a decrease in model performance. This suggests that our method

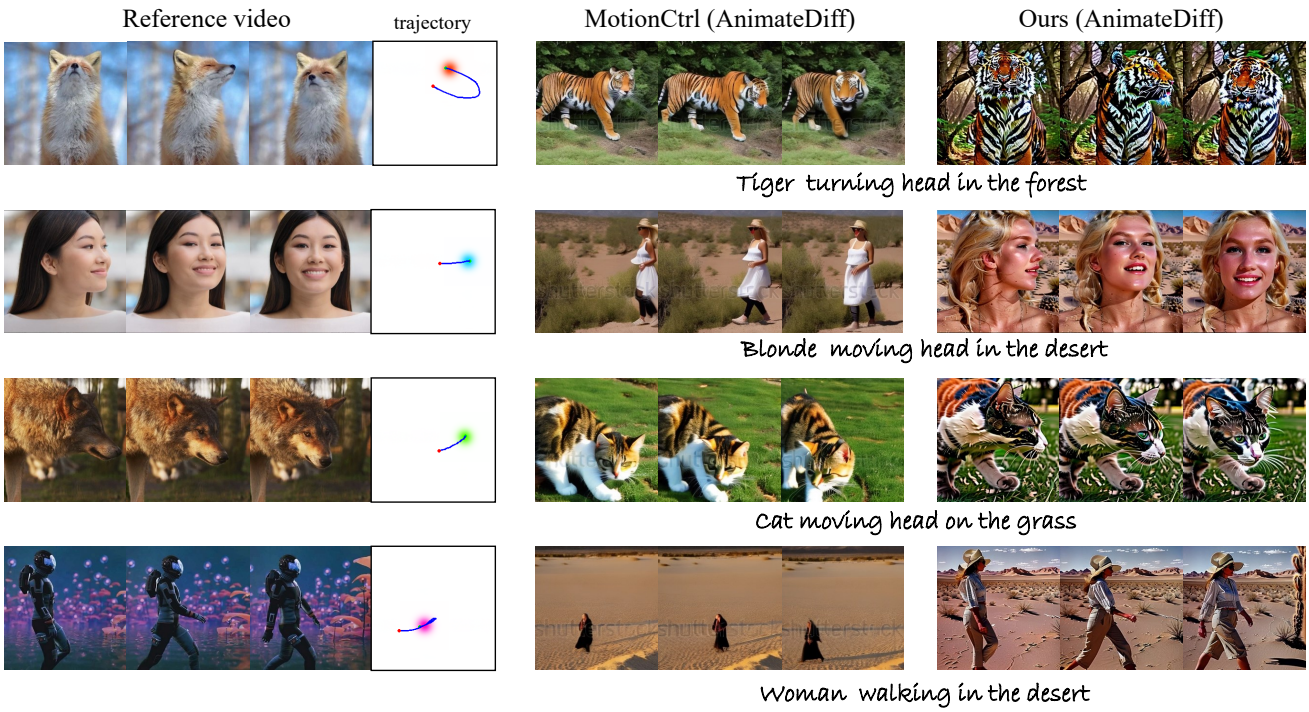


Figure 1. **Visualization results compared with trajectory-based method.**

requires high-quality motion transfer data to help the model learn generalized motion patterns. Due to our effective and reasonable use of model priors, the model exhibited good performance with just over 100 samples. However, blindly increasing the dataset size with low-quality data leads to detrimental results.

## References

[1] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024. 1

[2] Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv e-prints*, pages arXiv–2305, 2023. 2

[3] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 1

[4] Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. Motionclone: Training-free motion cloning for controllable video generation. *arXiv preprint arXiv:2406.05338*, 2024. 2

[5] 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. 1

[6] Luozhou Wang, Ziyang Mai, Guibao Shen, Yixun Liang, Xin Tao, Pengfei Wan, Di Zhang, Yijun Li, and Yingcong Chen. Motion inversion for video customization. *arXiv preprint arXiv:2403.20193*, 2024. 1, 2

[7] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems*, 36:7594–7611, 2023. 2

[8] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[9] Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. *arXiv preprint arXiv:2405.14864*, 2024. 2

[10] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8466–8476, 2024. 1, 2

[11] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023. 2

[12] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jia-Wei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. In *European Conference on Computer Vision*, pages 273–290. Springer, 2024. 1, 2