MOTION INVERSION FOR VIDEO CUSTOMIZATION

Anonymous authors
Paper under double-blind review

A APPENDIX

A.1 MOTION EMBEDDINGS

Parameters Details. In the architecture of the video diffusion model's UNet, which includes both ZeroScope cerspense (2023) and ModelScope Wang et al. (2023), there is a total of 16 temporal transformers, which means $L = 16$. The model is divided into three main sections: the downsampling blocks, the mid block, and the upsampling blocks. Our model has a downsampling part with 6 blocks (2 with 320 channels, 2 with 640, and 2 with 1280 channels) and a mid block with 1280 channels. The upsampling part has 9 blocks (3 with 320 channels, 3 with 640, and 3 with 1280 channels). We added motion embeddings to each block in both downsampling and upsampling parts, totaling 15 embeddings. This approach works well for complex, wide-ranging motion. However, for simpler, localized motion, fewer embeddings (only in the 1280-channel blocks, totaling 5) proved more effective. In this case, the shape of 1D embedding for QK is $(1, N, 1280)$, while the 2D embedding for V is $(H/32 \times W/32, N, 1280)$, where N, H and W refer to the number of frames, height and width of source video respectively. This balance helps us keep the motion details accurate while being efficient.

We also introduce a scaling factor γ to control the strength of the motion value embedding, as shown in equation 1. Different scales are applied depending on the type of motion to achieve better results. For instance, we set the scale to 0 when handling simple, global camera movements. This removes local constraints, allowing the model to generalize better and improve text alignment.

$$\text{TA}_i(\mathbf{F}) = \text{softmax}\left(\frac{(\mathbf{W_q}(\mathbf{F}+\mathbf{m}_i^{QK}))(\mathbf{W_k}(\mathbf{F}+\mathbf{m}_i^{QK}))^T}{\sqrt{d_k}}\right)(\mathbf{W_v}(\mathbf{F}+\gamma * \mathbf{m}_i^{V})), \tag{1}$$

A.2 TRAINING OBJECTIVES

In this work, we discuss two additional training objectives beyond the conventional approach:

1. The first is from VMC Jeong et al. (2023), and the loss function is defined as:

$$\mathcal{L} = 1 - \mathbb{E}_t\left[\cos\left(\Delta\epsilon_\theta(x_t^{1:N}, t), \Delta\epsilon_t^{1:N}\right)\right] \tag{2}$$

   Here, Δ denotes the difference operation applied to consecutive frames, effectively capturing the temporal change or residual between them. Specifically: $\Delta\epsilon_\theta(x_t^{1:N}, t)$ represents the model's predicted residual, calculated as the absolute difference between the model's predictions for successive frames. $\Delta\epsilon_t^{1:N}$ stands for the target residual, computed as the absolute difference between the target frames' values. The cosine similarity $\cos$ between these residuals is then computed, and the mean value is subtracted from 1 to obtain the final loss $\mathcal{L}$, which is minimized during training.

2. The second loss function is from Zhao et al. (2023). This loss formulation is meticulously designed to minimize the influence of appearance features on temporal noise prediction. The appearance-debiased temporal loss is computed as follows: Given a noise vector ϵ_i, the adjustment for appearance bias is calculated using:

$$\phi(\epsilon_i) = \sqrt{\beta^2 + \epsilon_i^2 - \beta\epsilon_{anchor}}, \tag{3}$$

   where β is a tuning parameter that controls the degree of decentralization, and ϵ_{anchor} serves as a randomly chosen reference point among the frame noises.

The temporal loss, adjusted for appearance bias, is then:

$$L_{ad-temp} = \mathbb{E}_{z_0,t}\left[\|\phi(\epsilon) - \phi(\epsilon_\theta(z_t, t, \mathcal{M}))\|^2\right], \tag{4}$$

where z_t represents the latent codes, t the time step, and ∇y the gradient with respect to the data.

This debiased loss is integrated with the standard temporal loss, yielding the composite objective:

$$L_{temporal} = L_{org-temp} + L_{ad-temp}. \tag{5}$$

Here $L_{org-temp}$ represent the default MSE loss.

We noticed that different types of motion benefit from different loss functions. For camera motion, where the subject details from the source video become irrelevant, we use the second loss function to improve the motion embeddings' inversion. We found that using our method eliminates the need to learn a spatial LoRA for improved appearance-motion disentanglement. This not only reduces the complexity but also enhances the generalization ability of our approach. Recent video models (Yang et al., 2024; Lab & etc., 2024) no longer use a combination of 2D spatial and 1D temporal attention, opting instead for unified 3D attention. As a result, spatial LoRA is no longer applicable in such cases

Note that our approach primarily emphasize motion representation, and is orthogonal to that of MotionDirector Zhao et al. (2023) and VMC Jeong et al. (2023). It is this orthogonal nature that enables us to employ a comprehensive set of optimization objectives.

A.3 Implementation Details

Optimization Details. We adopt the AdamW optimizer, as introduced by Loshchilov & Hutter (2019), with key hyperparameters configured for our experiments. The learning rate is set to 5×10^{-2}. The exponential decay rates for the first and second moment estimates are 0.9 and 0.999, respectively. Weight decay regularization is applied at 1×10^{-2}, and the numerical stability constant ϵ is 1×10^{-8}. The learning rate for spatial LoRA is an order of magnitude lower at 5×10^{-4}. For consistency in evaluating our method, we fixed the number of iterations at 800 for both quantitative and qualitative analyses. However, we observed that for practical applications, specifically in modeling camera motion, fewer iterations, typically in the range of 100 to 300, are often adequate. It is noteworthy that the learning rate for spatial LoRA markedly impacts the convergence behavior of our training process.

Inference Details. In all our experiments, the resolution of video frames is set to either 512×320 or 400×400, striking a balance between computational efficiency and the level of detail necessary for effective motion representation. Classifier-free guidance is employed uniformly across all models, both baseline and proposed, with the guidance scale parameter set to 10. Generation is performed using the Denoising Diffusion Implicit Models (DDIM) scheduler with the total number of inference steps fixed at 50.

Resource Costs. In terms of **GPU memory usage**, our methodology is optimized for efficiency, necessitating less than 14GB of VRAM for the inversion of videos at a resolution of 512 x 320. This optimization renders our approach compatible with a broad spectrum of contemporary hardware setups. Regarding **computational time**, the training phase of our method is notably expedient, achieving approximately 800 iterations in merely 10 minutes when utilizing an NVIDIA 3090 GPU. This efficiency is particularly beneficial as it translates to no additional time requirements during the inference phase, thereby ensuring a seamless and efficient operational flow.

A.4 Noise Initialization

In the realm of motion transfer, the method of noise initialization is crucial for the final outcomes. Herein, we delineate the initialization strategies employed by various baseline methods and our chosen approach in the experiment. These strategies all leverage the inversion noise derived from the DDIM inversion of the source video:

- In VMC Jeong et al. (2023), the initial noise is directly taken from the DDIM inversion of the source video, preserving the original video's motion characteristics but limiting variation across different seeds.

- DMT Yatim et al. (2023) replaces low-frequency components of random noise with those from inversion noise. The low-frequency components are obtained through downsampling and upsampling operations by a factor of 4.
- MotionDirector Zhao et al. (2023) combines inversion noise with random noise, modulated by a scaling factor set to 0.5 in experiments.

Our approach mirrors the strategy of MotionDirector, appreciating the balanced amalgamation of authentic motion cues with the potential for varied expressions, achieved through the strategic mixing of inversion and random noises. This exploration of noise initialization techniques underscores the delicate balance between preserving the source video's inherent motion and injecting diversity for enriched motion transfer outcomes.

A.5 Validation Dataset

Our validation dataset consists of three types of motion, object, camera and hybrid. For object and hybrid motion, these validation video are simply adopted from DMT Yatim et al. (2023), and they are from DAVIS dataset Perazzi et al. (2016). For camera, we adopt some from github repo of MotionDirector Zhao et al. (2023) and online sources. Extact videos and prompts we used please see the attached files.

A.6 Results

To enhance understanding, we have included a more detailed set of images, carefully selecting 8 frames evenly distributed across the entire sequence of 16. Please note that we've corrected an error in the VMC results presented here and will update the main paper accordingly. For all video results presented in our main paper. Please see the attached files. We include comparisons with three video editing methods (Wu et al., 2023; Chen et al., 2023; Geyer et al., 2023) in Table. 1. However, it is important to note that **video editing and motion transfer are different tasks** and video editing does not allow drastic shape modifications.

Method	**Text Similarity** ↑	**Motion Fidelity** ↑
Tune-A-Video-ICCV23	0.2821	0.6604
Control-A-Video	0.2581	0.8720
TokenFlow-ICLR24	0.2680	**0.9735**
Ours	**0.3113**	0.9552

Table 1: Comparisons with video editing methods.

References

cerspense. zeroscope_v2. https://huggingface.co/cerspense/zeroscope_v2_576w, 2023. Accessed: 2023-02-03.

Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023.

Hyeonho Jeong, Geon Yeong Park, and Jong Chul Ye. Vmc: Video motion customization using temporal attention adaption for text-to-video diffusion models. *arXiv preprint arXiv:2312.00845*, 2023.

PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, April 2024. URL https://doi.org/10.5281/zenodo.10948109.

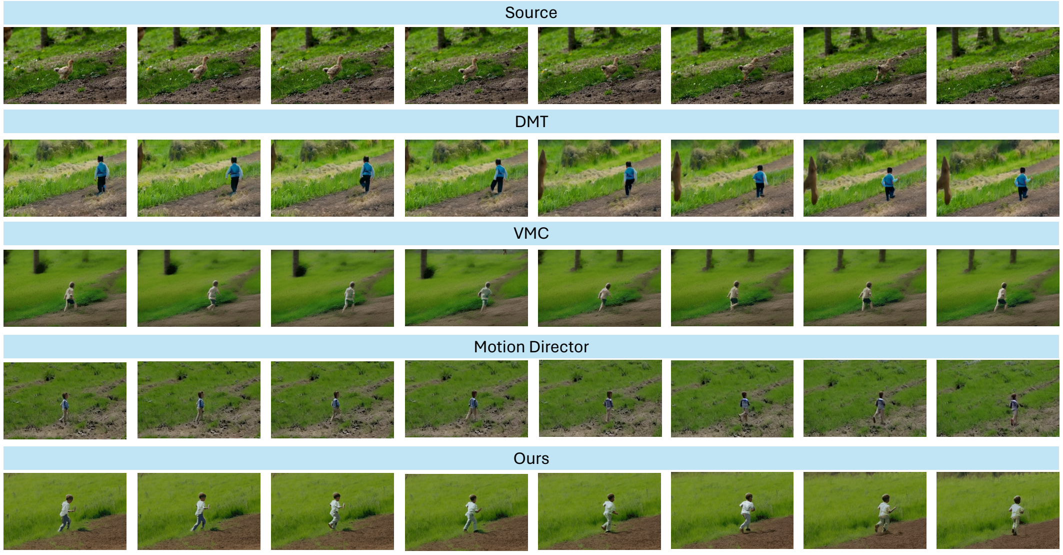

Figure 1: Additional Results: Showcasing 8 Frames Selected from a Total of 16

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 724–732, 2016.

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.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7623–7633, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Danah Yatim, Rafail Fridman, Omer Bar Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. *arXiv preprint arXiv:2311.17009*, 2023.

Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jiawei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. *arXiv preprint arXiv:2310.08465*, 2023.

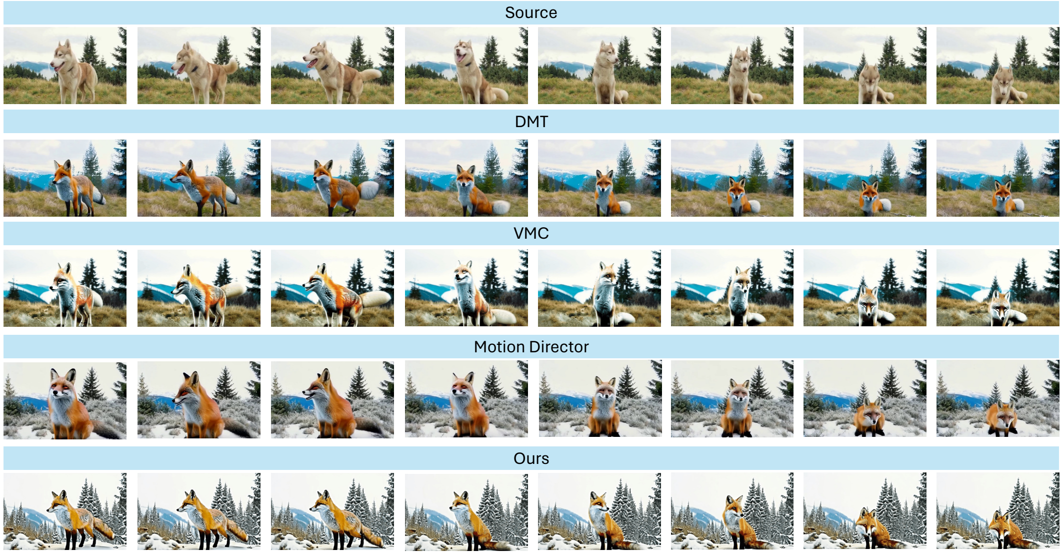

Figure 2: Additional Results: Showcasing 8 Frames Selected from a Total of 16

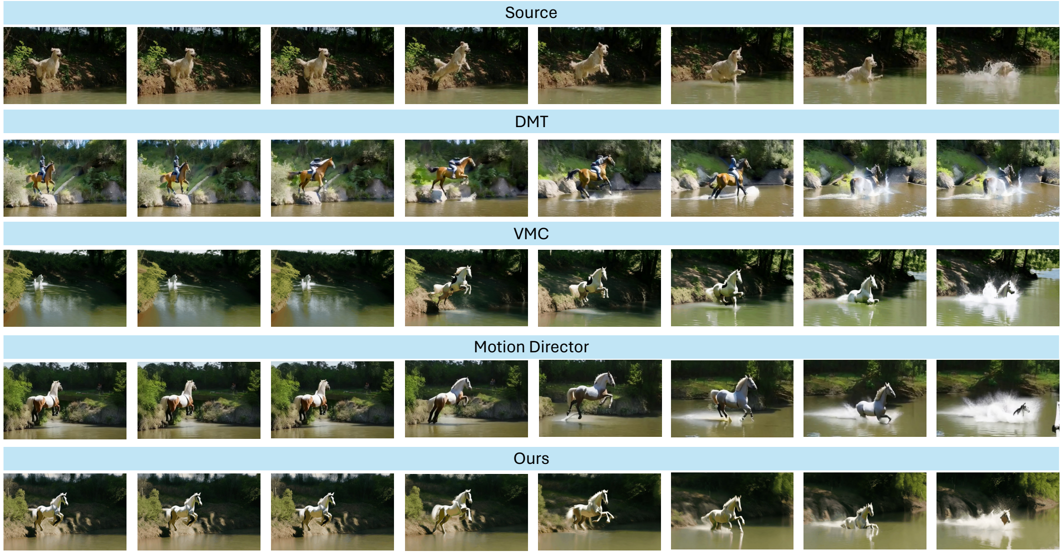

Figure 3: Additional Results: Showcasing 8 Frames Selected from a Total of 16

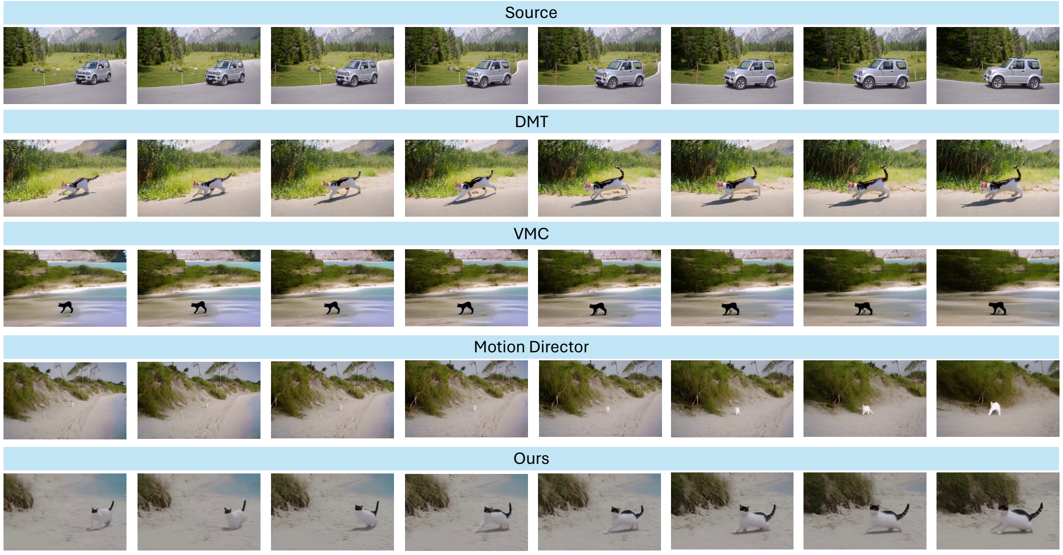

Figure 4: Additional Results: Showcasing 8 Frames Selected from a Total of 16

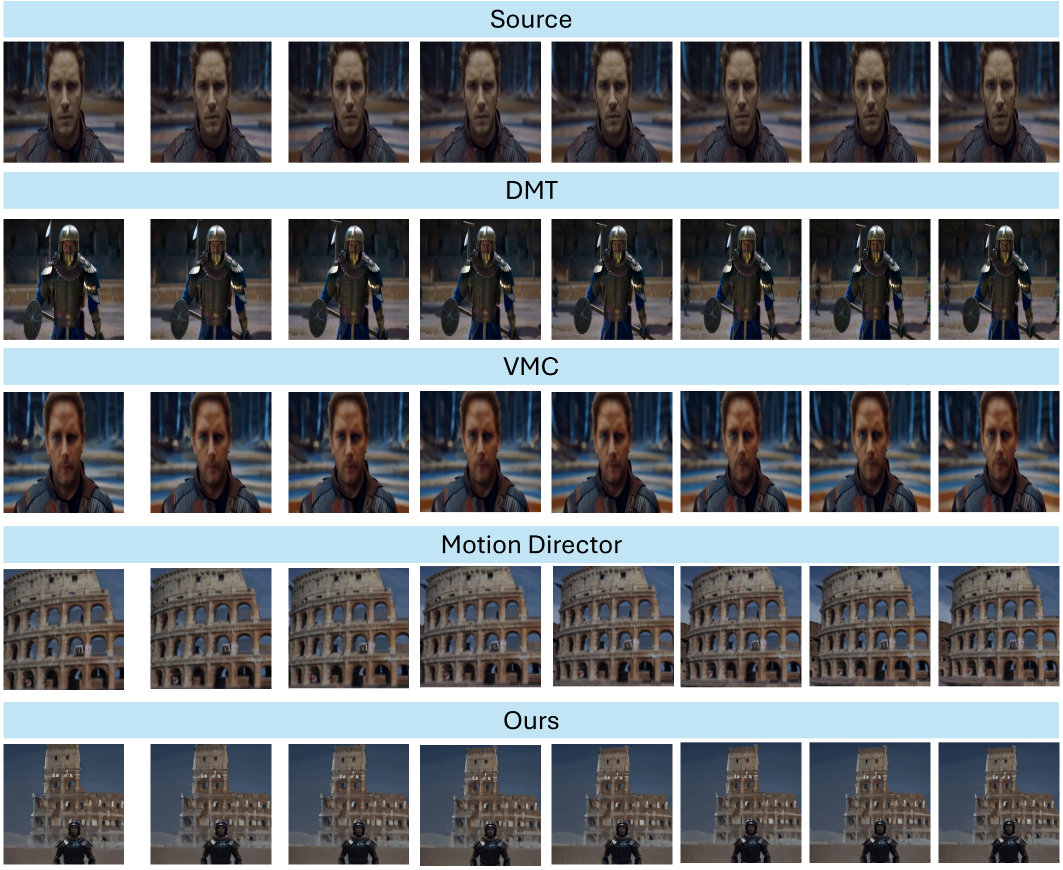

Figure 5: Additional Results: Showcasing 8 Frames Selected from a Total of 16