Supplement Material of Arbitrary Motion Style Transfer with Multi-condition Motion Latent Diffusion Model

Wenfeng Song[1], Xingliang Jin[1], Shuai Li[2,3*], Chenglizhao Chen[4†], Aimin Hao[3,5], Xia Hou[1], Ning Li[1], Hong Qin[6]

[1]Beijing Information Science and Technology University, China [2]Zhongguancun Laboratory, China
[3]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[4]College of Computer Science and Technology, China University of Petroleum (East China)
[5]Research Unit of Virtual Human and Virtual Surgery (2019RU004), Chinese Academy of Medical Sciences
[6]Department of Computer Science, Stony Brook University (SUNY at Stony Brook), Stony Brook, New York 11794-2424, USA

1. Overview

In this supplementary material, we first provide additional experiments to evaluate our Multi-condition Motion Latent Diffusion Model (MCM-LDM) in Sec. 2. Then, we provide a comprehensive explanation of our quantitative metrics in Sec. 3. We further introduce the details of our user study in Sec. 4. Lastly, we provide the details of our Multi-condition Extraction in Sec. 5.

2. Additional Experiments

2.1. Importance of Diffusion in Latent Space

In this section, we further investigate the impact of the diffusion process in the latent space (Table 1: 'Ours w Latent Space') and the original motion space (Table 1: 'Ours w Motion Space'). In motion space diffusion, we directly apply the diffusion process to the original motion features instead of the latent features. Results demonstrate that the motion space model excels in preserving motion content (CRA improves by 14.43) compared to the latent space model. However, there is a substantial decrease in style representation performance (SRA decreases by 8.34) and a slight decline in motion quality (FMD decreases by 1.91). These results demonstrate that the motion space model prioritizes content preservation, leading to suboptimal style transfer. Moreover, this model consumes significant computational resources, taking an average of 7.690 seconds to generate 60 frames of motion, while our latent space model only requires 0.220 seconds, making it nearly 35 times faster. Thus, we choose to implement our method in the latent space as it strikes a balance between style and content, significantly improving style transfer efficiency.

Methods	FMD↓	CRA↑ (%)	SRA↑ (%)	TSI↓
Ours w Motion Space	29.60	**50.18**	49.66	**0.36**
Ours w Latent Space	**27.69**	35.75	**58.00**	0.40

Table 1. **Comparison with diffusion in motion space.** The results show that our MCM-LDM implemented in latent space can better transfer the style.

2.2. More Guidance Strategies in Multi-condition Denoiser

As shown in Table 2, we further experiment with adding our content f_c, trajectory f_t, and style f_s to our Multi-condition Denoiser E_θ as primary or secondary conditions, resulting in four more guidance strategies. Note that the primary conditions guide the denoising process in E_θ by concatenating with noise late features. The secondary conditions guide the denoising process in E_θ by incorporating into our E_θ using AdaLN-Zero [7]. For example, we experiment to treat all conditions as primary conditions (Table 2: ID=1), or as secondary conditions (Table 2: ID=4), and find that all metrics show a decrease compared to our method (Table 2: ID=5). The results indicate that our method can achieve a balanced performance only when content f_c is treated as the primary condition, and trajectory f_t and style f_s are treated as the secondary conditions to guide the denoising process.

2.3. Classifier-free Parameters

In this section, we conduct experiments on the parameters of classifier-free diffusion guidance [5]. As shown in Table 3, we initially change the dropout ratio (p) of the training none-style model from 0.05 to 0.45 and observe an upward trend in the FMD, CRA, and TSI metrics, while the SRA metric shows a downward trend. We also experiment with adjusting the guidance scale (λ) from 4.5 to 1.0 and

---

*,† Corresponding authors

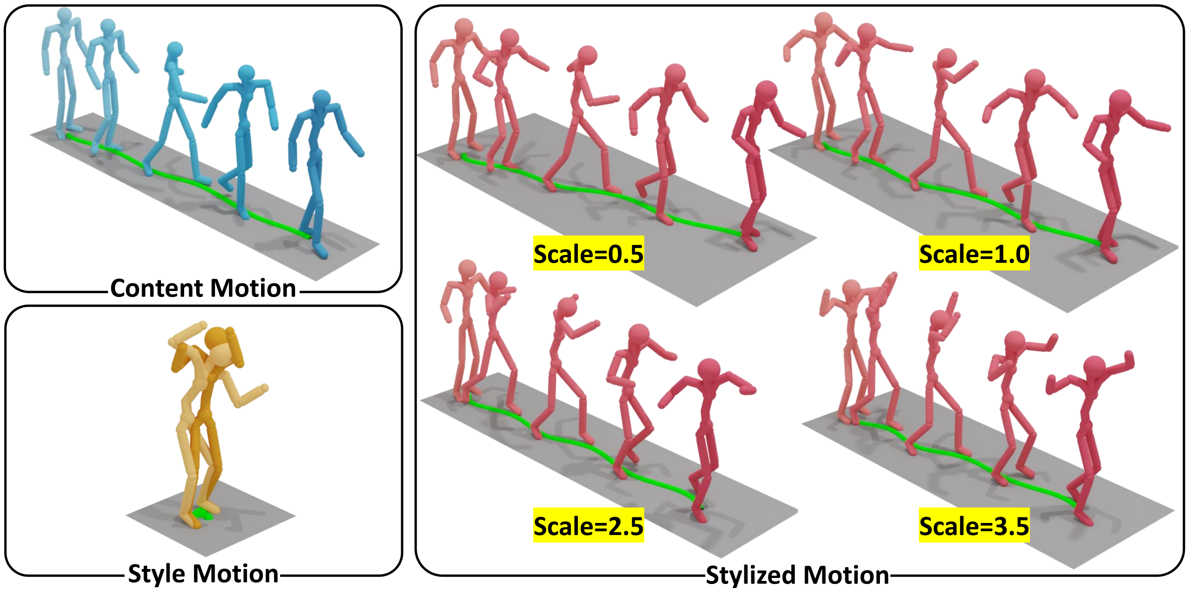


Figure 1. **Results of style control.** The results show that increasing the scale from 0.5 to 2.5 gradually strengthens the style, while further increases have minimal impact on style strength.

	Pri.	Sec.	FMD↓	CRA↑ (%)	SRA↑ (%)	TSI↓
1	f_c, f_t, f_s	–	46.71	22.00	55.55	0.66
2	f_c, f_t	f_s	33.44	29.50	**62.11**	0.58
3	f_c, f_s	f_t	29.65	33.56	59.33	0.45
4	–	f_c, f_s, f_t	38.74	32.87	43.22	0.49
5	f_c	f_s, f_t	**27.69**	**35.75**	58.00	**0.40**

Table 2. **Experiments of guidance strateges in E_θ.** 'Pri.' represents the primary conditions, while 'Sec.' represents the secondary conditions. When content f_c is treated as the primary condition, while trajectory f_t and style f_s are treated as the secondary conditions to guide the denoising process, our method achieves better performance.

observe the same trend. We further validate the control of style using classifier-free guidance through the selection of scales during inference, as shown in Fig. 1. The visualization supports the effectiveness of classifier-free guidance in achieving user-friendly style transfer. To maintain balanced performance, we ultimately choose p=0.25 and λ=2.5.

2.4. Importance of the MotionCLIP as Our Style Extractor

In this section, we further investigate the reasons behind our choice of MotionCLIP [11] over a simple pre-trained AutoEncoder as our Style Extractor. The training process of MotionCLIP involves aligning the motion latent space with the text and image CLIP [9] spaces through the use of text loss and image loss. In order to assess the importance of this alignment for our model, we retrain a MotionCLIP variant without text loss and image loss, which essentially functions as a basic AutoEncoder (AE). The results, as presented in Table 4, demonstrate a decrease in the SRA metric. This decrease indicates that MotionCLIP, which aligns with the Clip space of text and images, can more effectively extract style features, thereby enhancing the effectiveness of our method.

Classifer-free		FMD↓	CRA↑ (%)	SRA↑ (%)	TSI↓
Dropout	Scale				
$p = 0.05$	$\lambda = 2.5$	31.22	33.00	61.22	0.50
$p = 0.15$	$\lambda = 2.5$	30.93	32.43	**62.33**	0.45
$p = 0.25$	$\lambda = 2.5$	**27.69**	35.75	58.00	0.40
$p = 0.35$	$\lambda = 2.5$	29.73	**36.68**	55.88	0.40
$p = 0.45$	$\lambda = 2.5$	27.81	36.31	55.55	**0.37**
$p = 0.25$	$\lambda = 1.0$	32.61	**45.43**	42.11	**0.29**
$p = 0.25$	$\lambda = 1.5$	28.96	40.12	49.77	0.33
$p = 0.25$	$\lambda = 2.5$	**27.69**	35.75	58.00	0.40
$p = 0.25$	$\lambda = 3.5$	29.58	31.43	61.33	0.46
$p = 0.25$	λ= 4.5	30.50	28.56	**62.11**	0.51

Table 3. **Evaluation of the classifier-free parameters dropout p and scale λ.** For balanced performance, we ultimately choose p=0.25 and λ=2.5.

2.5. Evaluation of the Dimensionality Reduction in StyleRemover

In this section, we further experiment with three dimensionality reductions in our StyleRemover. We reduce the dimensionality of the content features from 7x256 to 6x256, 5x256, and 4x256, respectively. The dimensionality reduction aims to decrease the amount of information in the con-

Methods	FMD↓	CRA↑ (%)	SRA↑ (%)	TSI↓
Ours w AE	29.78	33.75	52.33	0.45
Ours w MotionCLIP	**27.69**	**35.75**	**58.00**	**0.40**

Table 4. **Comparison with AE as our Style Extractor.** The results show that using MotionCLIP as our Style Extractor enables better capturing of style features.

Dimensions	FMD↓	CRA↑ (%)	SRA↑ (%)	TSI↓
6x256	**27.69**	**35.75**	58.00	0.40
5x256	27.85	34.18	58.88	0.40
4x256	28.48	35.06	**59.33**	0.40

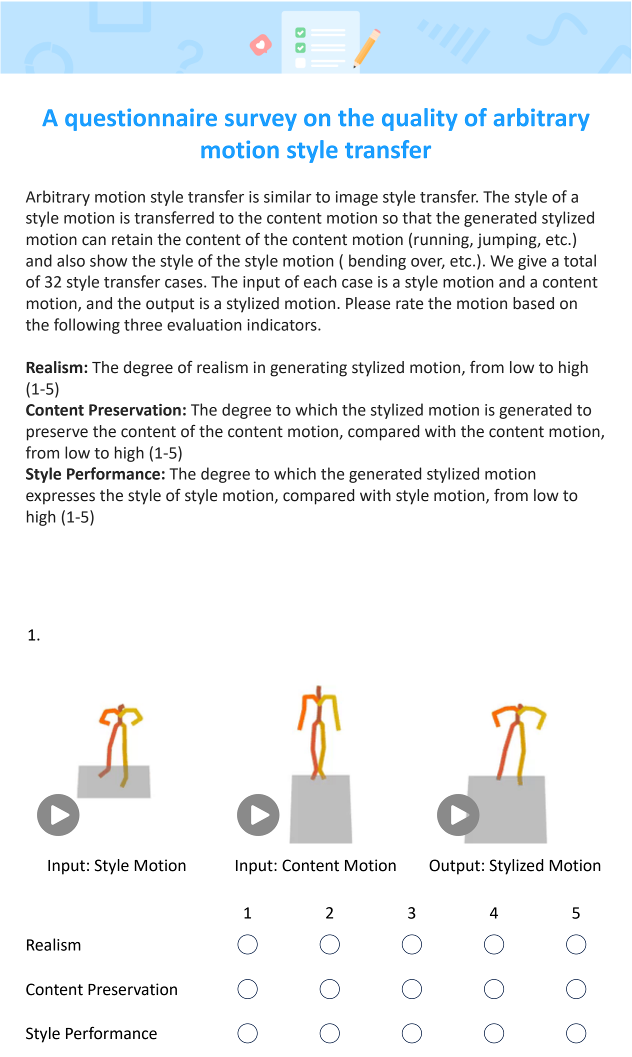
Table 5. **Evaluation of the dimensionality reduction in StyleRemover.** We experiment with reducing the dimensionality of the content features in StyleRemover from 7x256 to 6x256, 5x256, and 4x256, respectively. We reduce the dimensionality to 6x256, which achieves the best FMD metric, representing the highest overall motion quality.

tent features and achieve style removal. As shown in Table 5, as we reduce the dimensionality more, the style performance improves (SRA metric increasing), but at the cost of decreased content preservation and motion quality (CRA and FMD metric decreasing). To achieve a more balanced style transfer effect, we reduce the dimensionality to 6x256, which achieves the best FMD metric.

2.6. Evaluation of Sampling Strategy and Denoising Steps

As shown in Table 6, we experiment with DDIM and DDPM sampling strategy and different denoising steps of DDIM. The results show that DDIM yields better results than DDPM in our context. When using DDPM sampling, we notice a jitter in the stylized motions, which deteriorates all our metrics. The Content Recognition Accuracy (CRA) score falls from 35.75 to 21.50, and SRA drops from 58.00 to 26.55. This jitter might be attributed to error buildup in DDPM sampling. While both DDPM and DDIM are diffusion-based generative models, the key difference lies in sampling strategies. DDPM's continuous Markov sequence sampling can lead to error accumulation and potential quality degradation, whereas DDIM's skip-step approach mitigates this issue, resulting in higher-quality stylized motion generation.

Furthermore, the results of different DDIM denoising steps show that fewer steps (reducing 50 to 5) deteriorate the motion quality (with metrics like Fréchet Motion Distance (FMD) worsening from 27.69 to 29.46.). More steps (increasing to 500) slightly improve certain metrics like Style Recognition Accuracy (SRA) from 58.00 to 59.00 but at a higher computational cost. The choice of 50 steps represents our best balance between quality and efficiency.



A questionnaire survey on the quality of arbitrary motion style transfer

Arbitrary motion style transfer is similar to image style transfer. The style of a style motion is transferred to the content motion so that the generated stylized motion can retain the content of the content motion (running, jumping, etc.) and also show the style of the style motion (bending over, etc.). We give a total of 32 style transfer cases. The input of each case is a style motion and a content motion, and the output is a stylized motion. Please rate the motion based on the following three evaluation indicators.

Realism: The degree of realism in generating stylized motion, from low to high (1-5)
Content Preservation: The degree to which the stylized motion is generated to preserve the content of the content motion, compared with the content motion, from low to high (1-5)
Style Performance: The degree to which the generated stylized motion expresses the style of style motion, compared with style motion, from low to high (1-5)

1.

Input: Style Motion
Input: Content Motion
Output: Stylized Motion

	1	2	3	4	5
Realism	○	○	○	○	○
Content Preservation	○	○	○	○	○
Style Performance	○	○	○	○	○

Figure 2. **User interface in our user study.** Users need to rate each stylized motion on three metrics: Realism, Content Preservation, and Style Performance.

Sampling Strategy	Denoising Steps	FMD↓	CRA↑ (%)	SRA↑ (%)	TSI↓
DDIM	5	29.46	34.81	58.11	0.42
DDIM	50	27.69	**35.75**	58.00	**0.40**
DDIM	250	**27.37**	34.50	**59.55**	**0.40**
DDIM	500	27.99	34.25	59.00	**0.40**
DDPM	1000	145.05	21.50	26.55	0.80

Table 6. **Evaluation of sampling strategy and denoising steps.** The results show that DDIM sampling strategy is more compatible with our arbitrary motion style transfer task. We set the denoising steps to 50 to achieve a balance between effectiveness and computational efficiency.

In general, we chose the DDIM sampling strategy with 50 steps as it balances motion quality and computational efficiency.

Methods	FMD↓	CRA↑ (%)	SRA↑ (%)	FSF↓
Motion Puzzle [6]	166.36	24.36	**62.01**	1.41
Ours	**89.00**	**31.08**	41.11	**1.00**

Table 7. **Evaluation on Xia [12] test set.** The results show that our MCM-LDM outperforms Motion Puzzle in terms of FMD, CRA, and FSF metrics. Our lower performance of the SRA metric might be the inappropriate style encoder.

2.7. Experiments on CMU [2] and Xia [12] Datasets

To further test the generalizability of our MCM-LDM, we conduct experiments on motion data in BVH format. Specifically, we train our MCM-LDM on the CMU [2] dataset and test it on the Xia [12] dataset. Since BVH format data does not have an existing MotionCLIP as our Style Extractor, we use a pretrained VAE encoder. We also compared our MCM-LDM with Motion Puzzle [6], which uses the same dataset. We adhere to the quantitative and visual evaluation methodologies outlined in FineStyle [10].

As shown in Table 7, our MCM-LDM outperforms motion puzzle in terms of FMD and CRA, which proves the high quality of our generated motion and the high degree of content preservation. Moreover, our FSF metric is also higher than the Motion Puzzle, further validating the superiority of our MCM-LDM of treating trajectories as an additional learnable condition. This approach ensures more natural motion and avoids significant foot-sliding artifacts. Regarding the low performance on style metrics, we speculate that this is due to the VAE encoder's inability to encode style features effectively. It might be necessary to train a MotionCLIP equivalent with BVH format data, but this presents challenges with the current BVH datasets due to the lack of a large amount of annotated data.

In Fig. 3, we present the visual results, from which it can be observed that both our method and Motion Puzzle [6] successfully transfer the "Old" styles. However, our approach achieves a more complete leg-kicking motion, while Motion Puzzle fails to demonstrate a fully extended kick. Furthermore, our generated kicking motion firmly plants the standing leg on the ground, exhibiting a natural stance, whereas Motion Puzzle results in both legs kicking simultaneously, leading to an unnatural motion portrayal. The visual results further highlight our method's advantages in content preservation on the BVH dataset and in avoiding significant foot-sliding artifacts, ensuring a logical, scientifically sound, and fluid translation.

3. Details of Quantitative Metrics

In this section, we provide a detailed process for testing our five metrics, including Fréchet Motion Distance (FMD), Content Recognition Accuracy (CRA), Style Recognition Accuracy (SRA), Trajectory Similarity Index (TSI), and Foot Sliding Factor (FSF).

For FMD and CRA metrics, we need to train a content classifier first. Specifically, we obtain a labeled sub-dataset of CMU [2], CMU-8, based on the annotation information provided by Action2Motion [3]. This dataset encompasses eight categories of motions (including Walk, Wash, Run, Jump, Animal Behavior, Dance, Step, and Climb), totaling 1,186 motions. During testing, we select 40 motions from the HumanML3D [4] test set that are contained in CMU-8, and select five motions per category. We consider these motions as both content and style motion groups and the two groups of motions undergo mutual motion style transfer. The transfer process aims to simulate the task of arbitrary motion style transfer. The results in a total of 1600 stylized motions for each method. We compute the CRA for these stylized motions. Then, we regard them as fake motion, while CMU-8 serves as the real motion to determine the FMD value. We further test our TSI and FSF metrics using these stylized motions.

For the SRA metric, we first select six distinct long motions with diverse styles from the CMU [2] dataset, resulting in 130 motions after segmentation and mirroring. Using these motions, we train a style classifier. During testing, we choose 30 motions for both content and style motion groups for each method, resulting in 900 stylized motions through mutual style transfer. We then calculate the SRA metric for these stylized motions.

For the TSI metric, we calculate the distance between the stylized motion trajectory T_{gen} and the original content motion trajectory T_{con} using the Euclidean distance. The formula for the calculation is

$$TSI = \frac{1}{L}\sum_{i=1}^{L}\sqrt{(T_{con}^{i} - T_{gen}^{i})^2}, \tag{1}$$

where L represents the length of the generated motions; T_{gen}^{i} and T_{con}^{i} represent the trajectory feature for the i-th generated motion and the corresponding content motion respectively.

For the FSF metric, we first calculate the foot sliding displacement d_{gen}^{i} generated by the left and right feet during ground contact for each generated motion. We then add the foot sliding displacements of all movements to get the overall foot contact displacement d_{gen}. As the original content motions themselves exhibit some displacement while walking on the ground, we further calculate the overall foot sliding displacements d_{con} for the content motions. The final foot sliding factor for the generated motions can be obtained using

$$FSF = (d_{gen} - d_{con})/d_{con}. \tag{2}$$

This metric can effectively evaluate the degree of foot slip in generating motions.

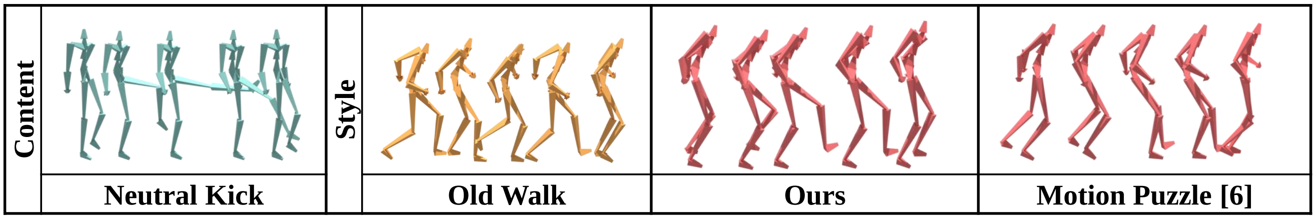


Figure 3. **Visual results on Xia [12] dataset.** Our MCM-LDM can better preserve the content with a fully extended leg.

4. User Study Details

In this section, we provide more details of our user study. We use the Wenjuanxing [1] website to design and collect our questionnaires. Our user study comprises 8 unique style transfer combinations. Each method is utilized to individually produce a stylized motion for each combination. This results in 8 unique stylized motions per method, all of which are then converted into a video format. Participants are asked to rate each result on a scale of 1 (significantly inaccurate) to 5 (significantly accurate), based on three metrics. Fig. 2 shows our designed user interface, where users should rate each stylized motion in the specific style transfer group.

As for user background, the study involves 40 participants of various backgrounds, including 25 students, 2 sales staff, 4 production workers, 3 teachers, and 6 individuals of other professions. Among them, there are 28 male users and 12 female users, including 2 under 18 years old, 28 between 18 and 25 years old, 6 between 26 and 30 years old, and 4 over 30 years old.

5. Architecture and Training Details of Our Multi-condition Extractor

In this section, we provide the architecture and training details of our Multi-condition Extractor.

5.1. Architecture

Our Multi-condition Extractor comprises three encoders: Style Extractor, Content Encoder, and Trajectory Encoder. The role of the Style Extractor is to extract the style features f_s of the motion $x^{1:L}$. To achieve this, we retrain the MotionCLIP [11] and utilize its encoder as our Style Extractor. Due to the alignment between the latent space of MotionCLIP and text/image, our Style Encoder can better capture the style features of the style motion.

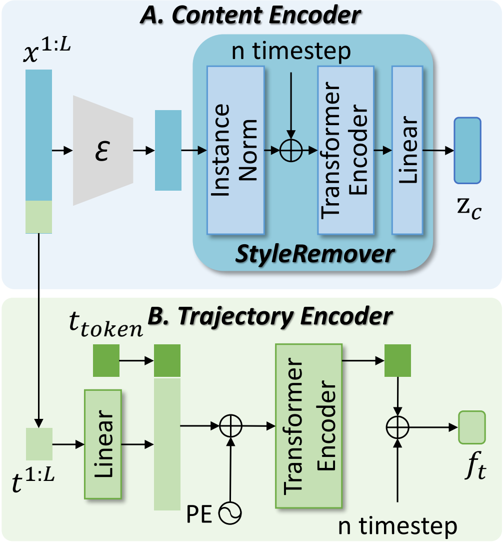


Figure 4. **Architecture of our Content Encoder and Trajectory Encoder.** We apply the StyleRemover module in our Content Encoder to strip style details from content motion $x^{1:L}$ and use a transformer network in the Trajectory Encoder to encode the motion trajectory $t^{1:L}$.

As shown in Fig. 4-A, our Content Encoder processes the content motion $x^{1:L}$. After initial processing by the pre-trained motion VAE encoder $\mathcal{E}$, we employ a StyleRemover module to eliminate the style information within the content. In StyleRemover, we use instance normalization (IN) and linear dimensionality reduction, preventing the model from overly depending on the content and thus avoiding style transfer failure.

For the Trajectory Encoder, as shown in Fig. 4-B, we draw inspiration from ACTOR [8] and employ a transformer-based encoder for trajectory encoding. Since the input motion lengths vary, we use a learnable parameter as the global trajectory token t_{token} and concatenate it with the projected trajectory sequence. The final trajectory condition feature is obtained by taking the corresponding dimension of the global trajectory token in the transformer output. This process can be summarized as:

$$f_t = TE\left(PE\left(\left[t_{token}, FC\left(t^{1:L}\right)\right]\right)\right)[0], \tag{3}$$

where $TE(\cdot)$ represent the Transformer Encoder; $PE(\cdot)$ is the Position Encoding; the $FC(\cdot)$ is the Linear layer.

5.2. Training Details

Except for the VAE encoder used in our Content Encoder and the MotionCLIP used for Style Extractor, which are pretrained, the other parts of our Multi-condition Extractor are involved in the final training of our denoising network. Within this, the transformer in the StyleRemover utilizes a 1-layer, 4-head configuration, while the transformer in the Trajectory Encoder uses a 2-layer, 4-head configuration. We have encoded all conditions into a 256-dimensional feature space.

References

[1] Wenjuanxing. `https://www.wjx.cn/`. 5

[2] CMU. Carnegie-mellon mocap database. `http://mocap.cs.cmu.edu/`. 4

[3] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the International Conference on Multimedia*, pages 2021–2029, 2020. 4

[4] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 4

[5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1

[6] Deok-Kyeong Jang, Soomin Park, and Sung-Hee Lee. Motion puzzle: Arbitrary motion style transfer by body part. *ACM Transactions on Graphics*, 41(3):1–16, 2022. 4

[7] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1

[8] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 5

[9] 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*, pages 8748–8763, 2021. 2

[10] Wenfeng Song, Xingliang Jin, Shuai Li, Chenglizhao Chen, Aimin Hao, and Xia Hou. Finestyle: Semantic-aware fine-grained motion style transfer with dual interactive-flow fusion. *IEEE Transactions on Visualization and Computer Graphics*, 29(11):4361–4371, 2023. 4

[11] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *Proceedings of the European Conference on Computer Vision*, pages 358–374, 2022. 2, 5

[12] Shihong Xia, Congyi Wang, Jinxiang Chai, and Jessica Hodgins. Realtime style transfer for unlabeled heterogeneous human motion. *ACM Transactions on Graphics*, 34(4):1–10, 2015. 4, 5