MotionFollower: Editing Video Motion via Score-Guided Diffusion

Supplementary Material

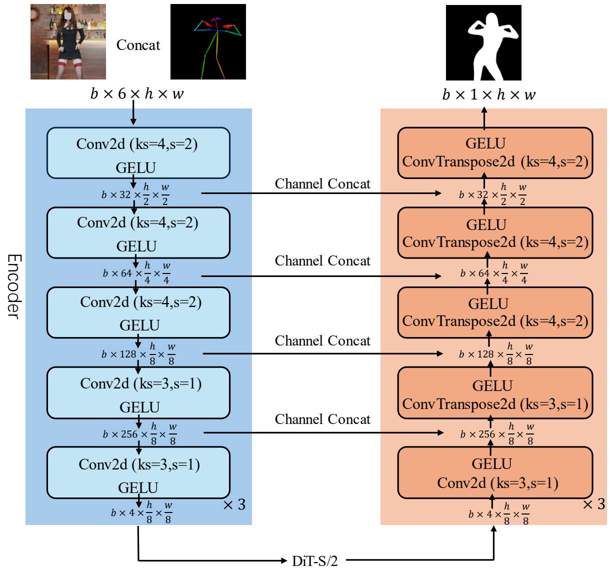
Concat
$b \times 6 \times h \times w$
$b \times 1 \times h \times w$
Encoder
Decoder
Conv2d (ks=4,s=2) GELU
Conv2d (ks=4,s=2) GELU
Conv2d (ks=4,s=2) GELU
Conv2d (ks=3,s=1) GELU
Conv2d (ks=3,s=1) GELU
× 3
GELU ConvTranspose2d (ks=4,s=2)
GELU ConvTranspose2d (ks=4,s=2)
GELU ConvTranspose2d (ks=4,s=2)
GELU ConvTranspose2d (ks=3,s=1)
GELU Conv2d (ks=3,s=1)
× 3
Channel Concat
Channel Concat
Channel Concat
Channel Concat
$b \times 32 \times \frac{h}{2} \times \frac{w}{2}$
$b \times 64 \times \frac{h}{4} \times \frac{w}{4}$
$b \times 128 \times \frac{h}{8} \times \frac{w}{8}$
$b \times 256 \times \frac{h}{8} \times \frac{w}{8}$
$b \times 4 \times \frac{h}{8} \times \frac{w}{8}$
DiT-S/2

Figure 1. The overview of our person segmentation model.

1. Person Segmentation

To gain the protagonist mask from the given condition, we propose a lightweight person segmentation model based on DiT [7] architecture, where the parameter number of our segmentation model is only 23.5MB compared with SAM-B [4] (91MB). The overview is shown in Figure. 1. SAM-B requires 40 seconds to extract masks of 24 frames, while our segmentation model only requires 6 seconds to accomplish the same job. Given a single image and a target pose, our segmentation model can predict the protagonist mask aligning with the target pose. Concretely, we concatenate the single image with the target pose in the channel dimension as the input of our model. We leverage several convolution blocks to extract different levels of representations, and the resulting features are sent to DiT for further modeling. The output of DiT is sent to the decoder to obtain the predicted mask, which is thresholded to obtain a binary mask. It is worth noting that the intermediate features from the first four convolution modules are concatenated with the outputs of the corresponding convolution modules in the decoder, which can contribute to enhancing controllable modeling.

In terms of training, we train our person segmentation model $\boldsymbol{Seg}(\cdot)$ using a 1000 video subset of our collected dataset. We train our segmentation model from scratch at the image level. With an average video length of 60 seconds and 30 FPS, the total number of training images exceeds 1.8 M. We split the entire video into two clips, where the first clip serves as the source and the second clip serves as the target. We obtain the poses and the masks of the protagonist from the target by employing DWPose [12] and SAM [4] on the target frames. The source frame $\boldsymbol{F}_{sr}$ and the extracted pose $\boldsymbol{P}_{tg}$ from the target serve as the inputs of our person segmentation model during training, and the extracted mask $\boldsymbol{M}_{tg}$ from the target serves as ground truth. We implement the MSE loss to train our model as follows:

$$\begin{aligned} \boldsymbol{M}^{pred} &= \boldsymbol{Seg}(\boldsymbol{F}_{sr}, \boldsymbol{P}_{tg}), \\ \mathcal{L}_{mse}(\boldsymbol{M}^{pred}, \boldsymbol{M}_{tg}) &= \left\| \boldsymbol{M}^{pred} - \boldsymbol{M}_{tg} \right\|_2^2 . \end{aligned} \tag{1}$$

It is worth noting that all components of our person segmentation model are trainable.

2. Details of Controllers

The Pose Controller (PoCtr) consists of four convolution blocks with two convolution layers. The Reference Controller (ReCtr) includes four convolution blocks for downsampling source features to the same dimension as the diffusion latents. The detailed frameworks of PoCtr and ReCtr are shown in Figure. 2.

3. Preliminaries

Diffusion models [2, 8, 10] have shown gorgeous results for high-quality image synthesis. They are based on thermodynamics, consisting of a forward diffusion process and a reverse denoising process. During the forward process, models appended to a constant noise schedule $\boldsymbol{\alpha}_t$ add random noise to the source sample $\boldsymbol{x}_0$ at time step t for obtaining a noise sample $\boldsymbol{x}_t$:

$$\begin{aligned} \boldsymbol{q}(\boldsymbol{x}_{1:T}) &= \boldsymbol{q}(\boldsymbol{x}_0) \prod_{t=1}^{T} \boldsymbol{q}(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}), \\ \boldsymbol{q}(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}) &= \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t} \boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I}). \end{aligned} \tag{2}$$

The source sample $\boldsymbol{x}_0$ is ultimately inverted into Gaussian noise $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ after T forward steps. The reverse process recovers $\boldsymbol{x}_0$ from $\boldsymbol{x}_T$ by some denoising steps. The denoising network $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$ tends to predict the noise $\boldsymbol{\varepsilon}$ conditioned on the current sample $\boldsymbol{x}_t$ and time step t by training with a simplified mean squared error:

$$\mathcal{L}_{simple} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\varepsilon}, t}(\| \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t) \|^2). \tag{3}$$

Further, diffusion models can be regarded as continuous models [10]. According to Langevin dynamics [9], the continuous denoising process can be depicted as the score function $\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t)$, sampling from the Gaussian noise. Regrading the condition $\boldsymbol{c}$, the score function can be described as $\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t, \boldsymbol{c})$, supporting conditional denoising.

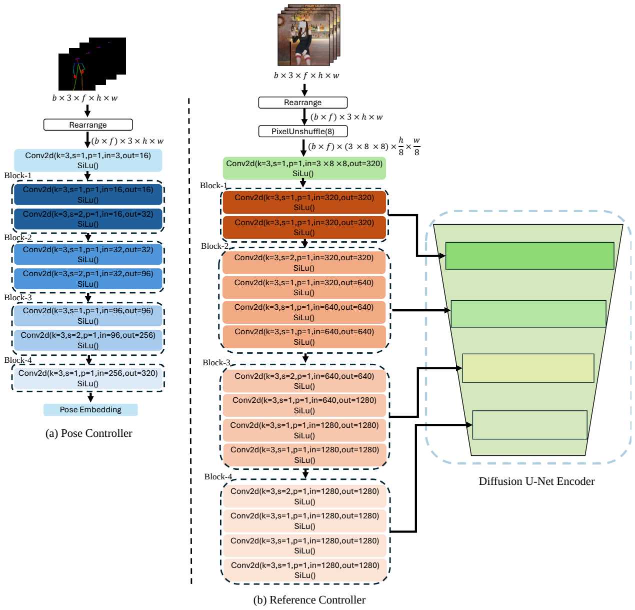
$b \times 3 \times f \times h \times w$
Rearrange
$(b \times f) \times 3 \times h \times w$
Conv2d(k=3,s=1,p=1,in=3,out=16) SiLu()
Block-1
Conv2d(k=3,s=1,p=1,in=16,out=16) SiLu()
Conv2d(k=3,s=2,p=1,in=16,out=32) SiLu()
Block-2
Conv2d(k=3,s=1,p=1,in=32,out=32) SiLu()
Conv2d(k=3,s=2,p=1,in=32,out=96) SiLu()
Block-3
Conv2d(k=3,s=1,p=1,in=96,out=96) SiLu()
Conv2d(k=3,s=2,p=1,in=96,out=256) SiLu()
Block-4
Conv2d(k=3,s=1,p=1,in=256,out=320) SiLu()
Pose Embedding
(a) Pose Controller
$b \times 3 \times f \times h \times w$
Rearrange
$(b \times f) \times 3 \times h \times w$
PixelUnshuffle(8)
$(b \times f) \times (3 \times 8 \times 8) \times \frac{h}{8} \times \frac{w}{8}$
Conv2d(k=3,s=1,p=1,in=3 ×8 ×8,out=320) SiLu()
Block-1
Conv2d(k=3,s=1,p=1,in=320,out=320) SiLu()
Conv2d(k=3,s=1,p=1,in=320,out=320) SiLu()
Block-2
Conv2d(k=3,s=2,p=1,in=320,out=320) SiLu()
Conv2d(k=3,s=1,p=1,in=320,out=640) SiLu()
Conv2d(k=3,s=1,p=1,in=640,out=640) SiLu()
Conv2d(k=3,s=1,p=1,in=640,out=640) SiLu()
Block-3
Conv2d(k=3,s=2,p=1,in=640,out=640) SiLu()
Conv2d(k=3,s=1,p=1,in=640,out=1280) SiLu()
Conv2d(k=3,s=1,p=1,in=1280,out=1280) SiLu()
Conv2d(k=3,s=1,p=1,in=1280,out=1280) SiLu()
Block-4
Conv2d(k=3,s=2,p=1,in=1280,out=1280) SiLu()
Conv2d(k=3,s=1,p=1,in=1280,out=1280) SiLu()
Conv2d(k=3,s=1,p=1,in=1280,out=1280) SiLu()
Conv2d(k=3,s=1,p=1,in=1280,out=1280) SiLu()
(b) Reference Controller
Diffusion U-Net Encoder

Figure 2. The overview of our PoCtr and ReCtr.

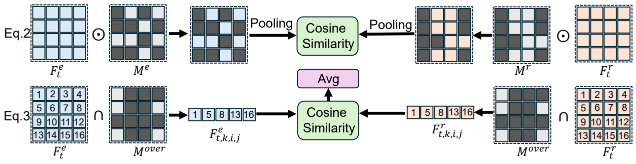

Figure 3. The pipeline of Eq.2 and Eq.3 of the main paper.

4. More Pooling Details

In Eq. 2 and Eq. 5 of the main paper, the elements ($\boldsymbol{F}_t^e \odot \boldsymbol{M}^e$, $\boldsymbol{F}_t^r \odot \boldsymbol{M}^r$, $\boldsymbol{F}_t^e \odot \boldsymbol{M}^{body}$, and $\boldsymbol{F}_t^r \odot (1 - \boldsymbol{M}^r)$) in `Pool`$(\cdot)$ have dimensions (T, N, H, W). These need to be pooled into a sequence for subsequent cosine similarity calculation. In Eq. 3 and Eq. 4 of the main paper, $\boldsymbol{F}_{t,k,i,j}^e$ and $\boldsymbol{F}_{t,k,i,j}^r$ are already sequences. For example, $\boldsymbol{F}_{t,k,i,j}^e$ represents a sequence where each element corresponds to a position with value=1 in $\boldsymbol{M}_{t,k,i,j}^{over}$ or $\boldsymbol{M}_{t,k,i,j}^{body}$. Figure.3 illustrates the pipeline of Eq.2 and Eq.3 for clarity.

5. Implementation of Consistency Guidance

According to previous works [1, 10], when the diffusion model $\boldsymbol{\varepsilon}_\theta(\boldsymbol{z}_t^e)$ tends to predict the noise added to the original frames, it can be converted to the form of the score function, which can be depicted as:

$$\nabla_{\boldsymbol{z}_t^e} \log q(\boldsymbol{z}_t^e) = -\frac{1}{\sqrt{1 - \bar{\boldsymbol{\alpha}}_t}} \boldsymbol{\varepsilon}_\theta(\boldsymbol{z}_t^e). \tag{4}$$

We implement the additional conditions $\boldsymbol{F}_t^e$ and $\boldsymbol{F}_t^r$ to the score function, which can be described as:

$$\begin{aligned} \nabla_{\boldsymbol{z}_t^e} \log q(\boldsymbol{z}_t^e, \boldsymbol{F}_t^e, \boldsymbol{F}_t^r) &= \nabla_{\boldsymbol{z}_t^e} \log q(\boldsymbol{z}_t^e) + \nabla_{\boldsymbol{z}_t^e} \log q(\boldsymbol{F}_t^e, \boldsymbol{F}_t^r | \boldsymbol{z}_t^e) \\ &= -\frac{1}{\sqrt{1 - \bar{\boldsymbol{\alpha}}_t}} \boldsymbol{\varepsilon}_\theta(\boldsymbol{z}_t^e) + \nabla_{\boldsymbol{z}_t^e} \log q(\boldsymbol{F}_t^e, \boldsymbol{F}_t^r | \boldsymbol{z}_t^e). \end{aligned} \tag{5}$$

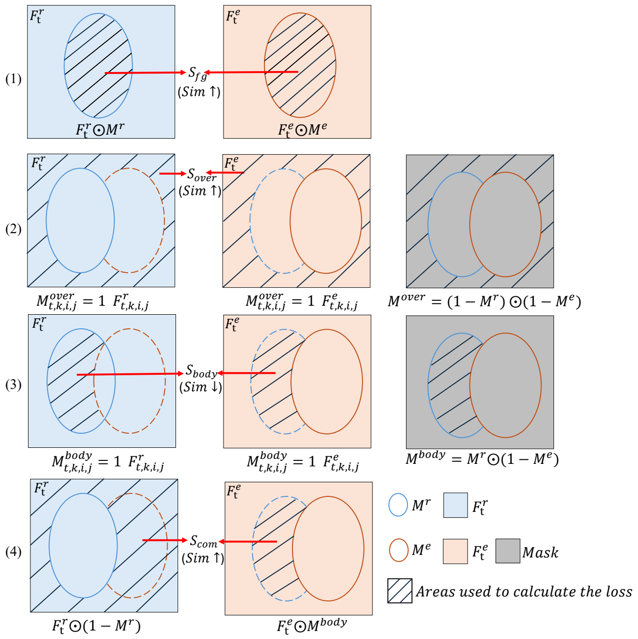

Figure 4. The illustration of our proposed functions.

We can ultimately obtain a new noise prediction model $\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{z}_t^e, \boldsymbol{F}_t^e, \boldsymbol{F}_t^r)$ for the joint distribution as follows:

$$\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{z}_t^e, \boldsymbol{F}_t^e, \boldsymbol{F}_t^r) = \boldsymbol{\varepsilon}_\theta(\boldsymbol{z}_t^e) - \sqrt{1 - \bar{\boldsymbol{\alpha}}_t} \nabla_{\boldsymbol{z}_t^e} \log q(\boldsymbol{F}_t^e, \boldsymbol{F}_t^r | \boldsymbol{z}_t^e). \tag{6}$$

Therefore, the sampling process equipped with our proposed consistency guidance can be depicted in Algorithm 1. `guidance`$(\cdot)$ and `decoder`$(\cdot)$ indicate our proposed consistency guidance and diffusion decoder. α_{fg}, α_{over}, α_{body}, α_{com} are set to 4.0, 6.0, 2.4, 1.2.

Furthermore, we illustrate our proposed 4 functions in Figure 4. The areas with black stripes are used to calculate the function. $Sim \uparrow$ indicates that the corresponding function encourages the two areas to have higher similarity and thus a similar appearance, while $Sim \downarrow$ means that the function aims to push the feature of these two areas away to avoid ghosting artifacts.

6. Advantages over MotionEditor

As MotionEditor is the only open-source video motion editing model, we set MotionEditor as our primary competitor in our comparison experiments. MotionFollower and MotionEditor are distinct in various aspects. MotionFollower has the following differences: (1) MotionEditor implements a resource-intensive attention injection mechanism to preserve the appearance of the source, while our MotionFollower novelly introduces score regularization to maintain consistency. Concretely, the attention injection mechanism of MotionEditor combines the keys and values from both the reconstructing and editing branches, thereby expanding the dimensions of each key and value. In contrast, MotionFollower abandons the previous consistency

Algorithm 1 Sampling process equipped with our consistency guidance

```
Input: Source video $\boldsymbol{V}_{sr}$, Source Mask $\boldsymbol{M}_r$, Source Pose $\boldsymbol{P}_{sr}$; Target Pose $\boldsymbol{P}_{tg}$, Predicted Mask $\boldsymbol{M}_e$
$\boldsymbol{z}_T^e \leftarrow$ sample from $\mathcal{N}(\mathbf{0}, \mathbf{I})$
for all $t$ from $T$ to $1$ do
    $\boldsymbol{\varepsilon}, \boldsymbol{F}_t^e, \boldsymbol{F}_t^r \leftarrow \boldsymbol{\varepsilon}_\theta(\boldsymbol{z}_t^e, \boldsymbol{V}_{sr}, \boldsymbol{P}_{sr}, \boldsymbol{P}_{tg})$
    $\nabla_{\boldsymbol{z}_t^e} \log q(\boldsymbol{F}_t^e, \boldsymbol{F}_t^r | \boldsymbol{z}_t^e) \leftarrow \texttt{guidance}(\boldsymbol{z}_t^e, \boldsymbol{F}_t^e, \boldsymbol{F}_t^r, \boldsymbol{M}_r, \boldsymbol{M}_e)$
    $\hat{\boldsymbol{\varepsilon}} \leftarrow \boldsymbol{\varepsilon} - \sqrt{1 - \bar{\boldsymbol{\alpha}}_t} \nabla_{\boldsymbol{z}_t^e} \log q(\boldsymbol{F}_t^e, \boldsymbol{F}_t^r | \boldsymbol{z}_t^e)$
    $\boldsymbol{z}_{t-1}^e \leftarrow \sqrt{\bar{\boldsymbol{\alpha}}_{t-1}} (\frac{\boldsymbol{z}_t^e - \sqrt{1-\bar{\boldsymbol{\alpha}}_t} \hat{\boldsymbol{\varepsilon}}}{\sqrt{\bar{\boldsymbol{\alpha}}_t}}) + \sqrt{1 - \bar{\boldsymbol{\alpha}}_{t-1}} \hat{\boldsymbol{\varepsilon}}$
end for
$\boldsymbol{x}_0 \leftarrow \texttt{decoder}(\boldsymbol{z}_0^e)$
return $\boldsymbol{x}_0$
```

preservation approaches, such as attention injection. We decouple the original discrete diffusion process into a continuous process which can be described as the score function, following the SDE principle. This score function steers the denoising towards a specific direction. Therefore, we propose multiple score regularization functions to guide the denoising process in the most appropriate direction, ensuring the most consistent results. To the best of our knowledge, MotionFollower is the first video editing model to explore score regularization. (2) MotionEditor only has an attention-based motion adaptor for supporting pose sequences, while MotionFollower has two relatively lightweight controllers (Pose Controller and Reference Controller) for modeling pose sequences and source videos. (3) MotionEditor is a one-shot video motion editing model that needs to be trained on each test video. In contrast, MotionFollower is trained on a video dataset and can be applied to arbitrary test videos without training.

Moreover, MotionFollower has the following advantages: (1) Our MotionFollower is capable of manipulating the video's motion while maintaining other extraneous details, such as large-scale camera movements, per-frame background variations, and the complex protagonist's appearance. By contrast, MotionEditor fails to handle particular videos featuring such scenarios. (2) MotionFollower is significantly lighter than MotionEditor. MotionFollower leverages score regularization to maintain consistency, rather than conventional attention injection, which expands the dimension of keys and values. (3) The inference speed of MotionFollower is significantly faster than MotionEditor. MotionFollower has a video processing throughput of 28.8 frames per minute, whereas MotionEditor processes at 1.6 frames per minute.

7. Framework Discussion

7.1. PosCtr

PosCtr is initialized with Gaussian weights and zero convolution is applied in the final projection layer. Thus, the pose features are Gaussian, and the sum of two Gaussian distributions remains Gaussian. Therefore, it makes sense to directly add pose feature to the diffusion latents.

7.2. ReCtr

We directly add features of ReCtr to the diffusion latents, as the diffusion model is capable of adaptively capturing appearance features. Therefore, when the protagonist's position in the source video is slightly misaligned with that in the target video, it does not impede the diffusion learning, as evidenced by the results in Sec. 10.

7.3. Dual Branches at Inference

We build up dual branches in inference for conducting our consistency guidance via score regularization, which may slightly increase the GPU memory consumption compared with the single branch-based architecture. Another simple solution is to replace our score regulation and dual branches with a simple ControlNet to preserve the appearance details and maintain content consistency. However, Table. 4 in the main paper shows that w/o score regulation and ours have similar GPU memory usage while replacing ControlNet with our controllers saves significant memory. Our score regulation ensures video consistency, particularly in scenes with large camera movement. However, a simple controller fails to preserve dynamic details in such scenarios, as evidenced by the results in Table. 4 and Figure. 5 in the main paper. Thus, it is necessary to remain the dual branch-based architecture to perform score regularization.

8. Predicted Masks

Figure. 5 illustrates the results of predicted masks by our person segmentation model. We can observe that our person segmentation model can accurately predict the masked regions based on the given pose and the body size of the protagonist in the given reference image.

9. Application Discussion

9.1. Multiple Motion Types

In addition to dancing, we conducted a qualitative experiment that considered other types of motion, such as run-

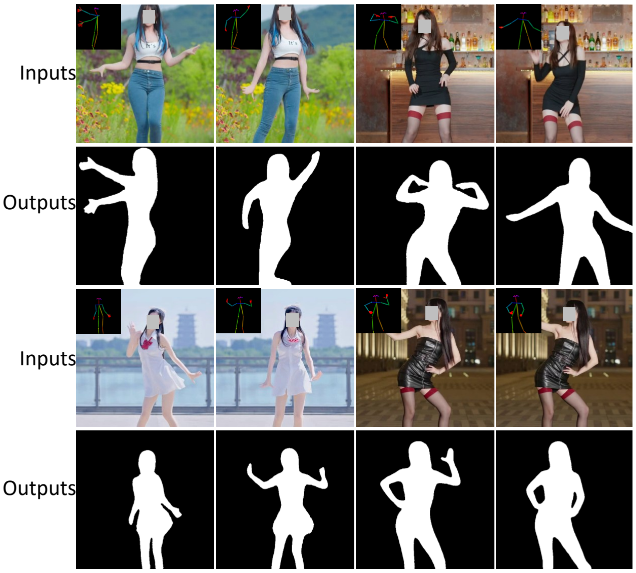

Figure 5. The results of predicted masks.

ning, playing Tai Chi, and playing basketball. The results are illustrated in Figure. 9. It is noticeable that our MotionFollower can handle various types of motion information.

9.2. Long Video Synthesis

Figure. 10 compares our MotionFollower and the strongest competitor MotionEditor on a long video of 600 frames, containing complex appearances and camera movements.

9.3. Camera Movement

Figure. 11 demonstrates the comparisons in a video that contains large camera movements. We modify Follow-Your-Pose [5] by replacing noise inputs with null-text optimized source frames [6] for editing.

9.4. Human Motion Transfer

Figure. 12 compares our method and other approaches to the Human Motion Transfer task. We also conduct an additional experiment on Human Motion Transfer, using the same case reported in previous human motion transfer papers [3, 11, 14], as shown in Figure. 13.

9.5. Comparison with Video Inpainting

The video motion editing can also be accomplished by employing video inpainting models (e.g. ProPainter [13]) and human motion transfer methods. However, the overall quality of this composed approach is not optimal. Figure. 6 demonstrates that our MotionFollower can drastically retain the background details and content consistency, while ProPainter [13]+Champ [14] fail to preserve the clothing/background details and suffer from blurry noises.

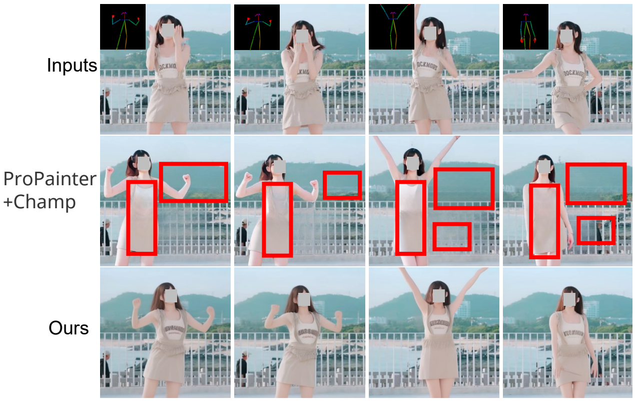

Figure 6. The comparison results between MotionFollower and ProPainter+Champ.

9.6. Videos with Orientation Changes

For some videos where the orientation changes significantly (e.g., from frontal to backward), $\mathcal{S}_{fg}(\cdot)$ has limitations, as it forces features in the predicted mask to align with those in the source mask. Thus, for this particular situation, we temporarily remove $\mathcal{S}_{fg}(\cdot)$ while keeping the rest unchanged, as shown in Figure. 8. Combining different score regulation functions can handle various scenarios.

10. Additional Results

Figure. 14, Figure. 15, and Figure. 16 show additional video motion editing results of our proposed MotionFollower in the specific videos featuring complicated backgrounds. Figure. 17 illustrates video motion editing results of our proposed model in the videos including complex initial poses. Figure. 18 shows the editing results of our MotionFollower in the videos with camera movements. Additionally, Figure. 19 shows the additional performance comparison results between our MotionFollower and the most advanced video motion editing model MotionEditor.

11. Additional Ablation Study

We conduct a more qualitative ablation study, as illustrated in Figure. 20. We can see that *w/o* $\boldsymbol{S}_{fg}$ results in degrading quality of the foreground. The plausible reason is that the diffusion model struggles to preserve the details of the protagonist's appearance without explicit foreground-related guidance. *w/o* $\boldsymbol{S}_{over}$ contributes to the occurrence of blurry noises in the dynamic background. The main reason is that it is relatively difficult for the diffusion model to capture and model the dynamic background, as its data distribution frequently varies. *w/o* $\boldsymbol{S}_{body}$ and *w/o* $\boldsymbol{S}_{com}$ both cause some semantic distortion due to the interference from non-overlapping protagonist's parts. The results demonstrate that our functions can effectively promote the model to preserve the appearance of foregrounds and backgrounds.

We further conduct an ablation study on masks used in

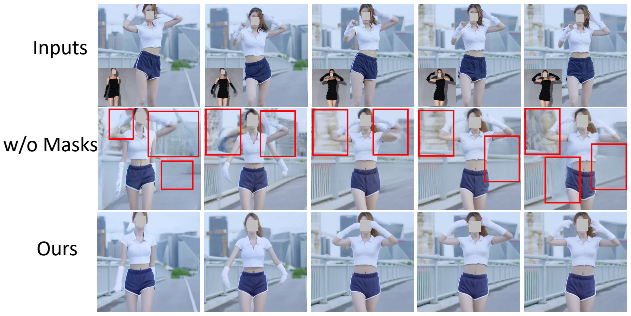

Figure 7. The ablation study on masks.

our score regulation, as shown in Figure. 7. w/o Masks refers to removing all masks used in the score regulation. The results indicate that MotionFollower can perform higher-quality video motion editing with leveraging multiple masks during our score regulation. Additionally, the results demonstrate that diffusion latents are highly abstracted compared to pixel space, but the predicted mask still makes sense in the feature space. Aggregated features in the spatial domain do not affect the model's perception of the spatial layout. This also can be validated by the fact that segmentation methods can still accurately predict masks on aggregated features, meaning the features well perceive the mask.

12. Limitations

Figure. 21 shows one failure case of our MotionFollower. The toy bear located behind the girl remains incomplete due to the foreground obstruction in the source video. Our model struggles to fill in the obscured areas in the background when encountering numerous small objects in the background. The probable solution is to explicitly introduce an additional inpainting adaptor to the diffusion model for recovering the background areas. This part is left as future work. Additionally, we find that it is challenging for our model to handle particular videos involving complex interactions with other objects. The plausible explanation is that interactions with other objects always result in occlusion. Our future work will focus on this challenging video.

13. Human Subjects Data Concern

Our training and testing datasets both involve human subjects, containing identifiable information. Since the collected videos are primarily from social media platforms (BiliBili, TikTok, and YouTube), we contacted the video uploaders via private messages to inquire whether they agreed to allow their videos to be used for non-commercial academic research, and we obtained their consent. All videos in our training and testing datasets have been approved by their corresponding video uploaders.

14. Ethical Concern

While MotionFollower has broad applicability, it is crucial to address several concerns: the risks of misuse in creating deceptive media, potential biases in training data, and the importance of respecting intellectual property.

References

[1] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 2

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1

[3] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *CVPR*, 2024. 4

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1

[5] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Ying Shan, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. In *AAAI*, 2024. 4

[6] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023. 4

[7] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *CVPR*, 2023. 1

[8] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1

[9] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *NeurIPS*, 2020. 1

[10] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 1, 2

[11] Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. In *CVPR*, 2024. 4

[12] Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *ICCV*, 2023. 1

[13] Shangchen Zhou, Chongyi Li, Kelvin C.K Chan, and Chen Change Loy. ProPainter: Improving propagation and transformer for video inpainting. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023. 4

[14] Shenhao Zhu, Junming Leo Chen, Zuozhuo Dai, Yinghui Xu, Xun Cao, Yao Yao, Hao Zhu, and Siyu Zhu. Champ: Controllable and consistent human image animation with 3d parametric guidance. In *EECV*, 2024. 4, 8

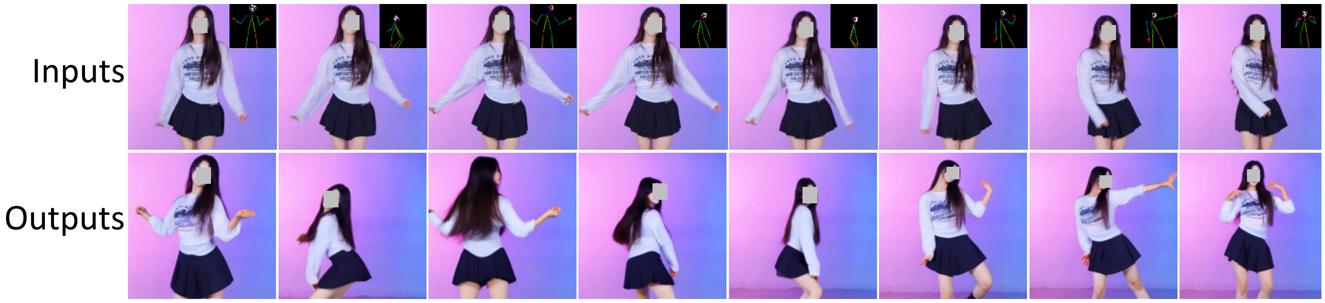

Figure 8. The edited video feature dramatic orientation variation.

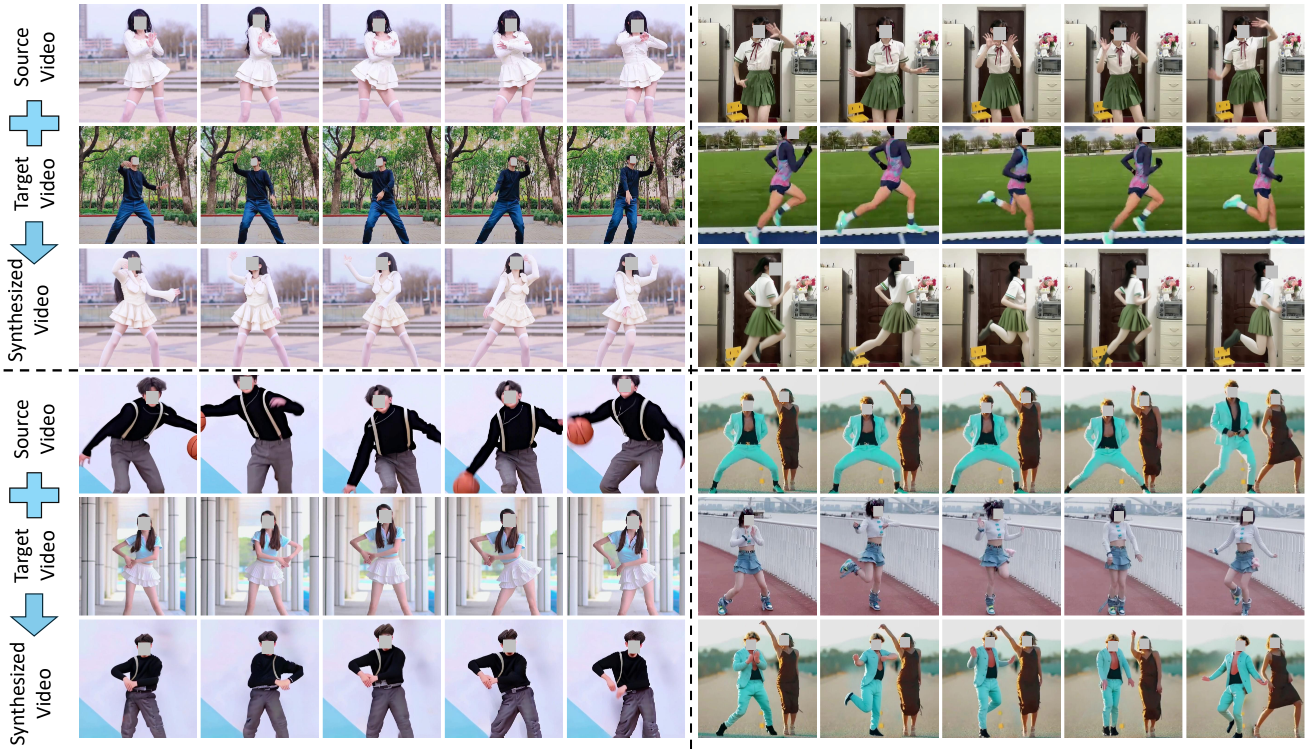

Figure 9. The edited videos feature various types of motion information.

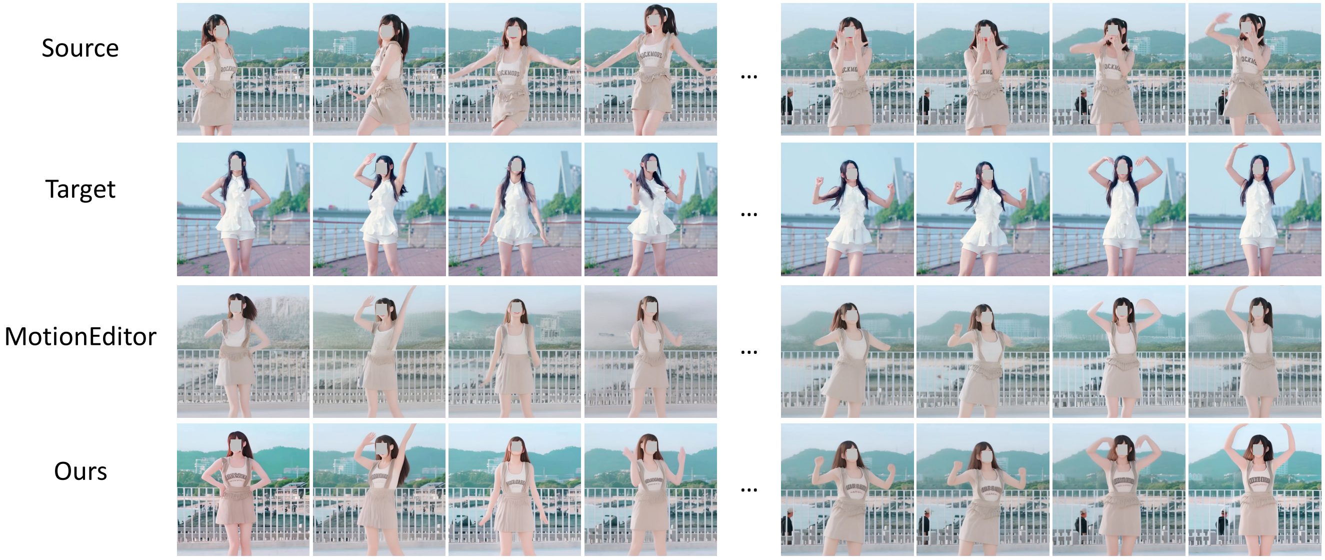

Figure 10. Qualitative comparison on a long video containing complicated appearances and camera movements.

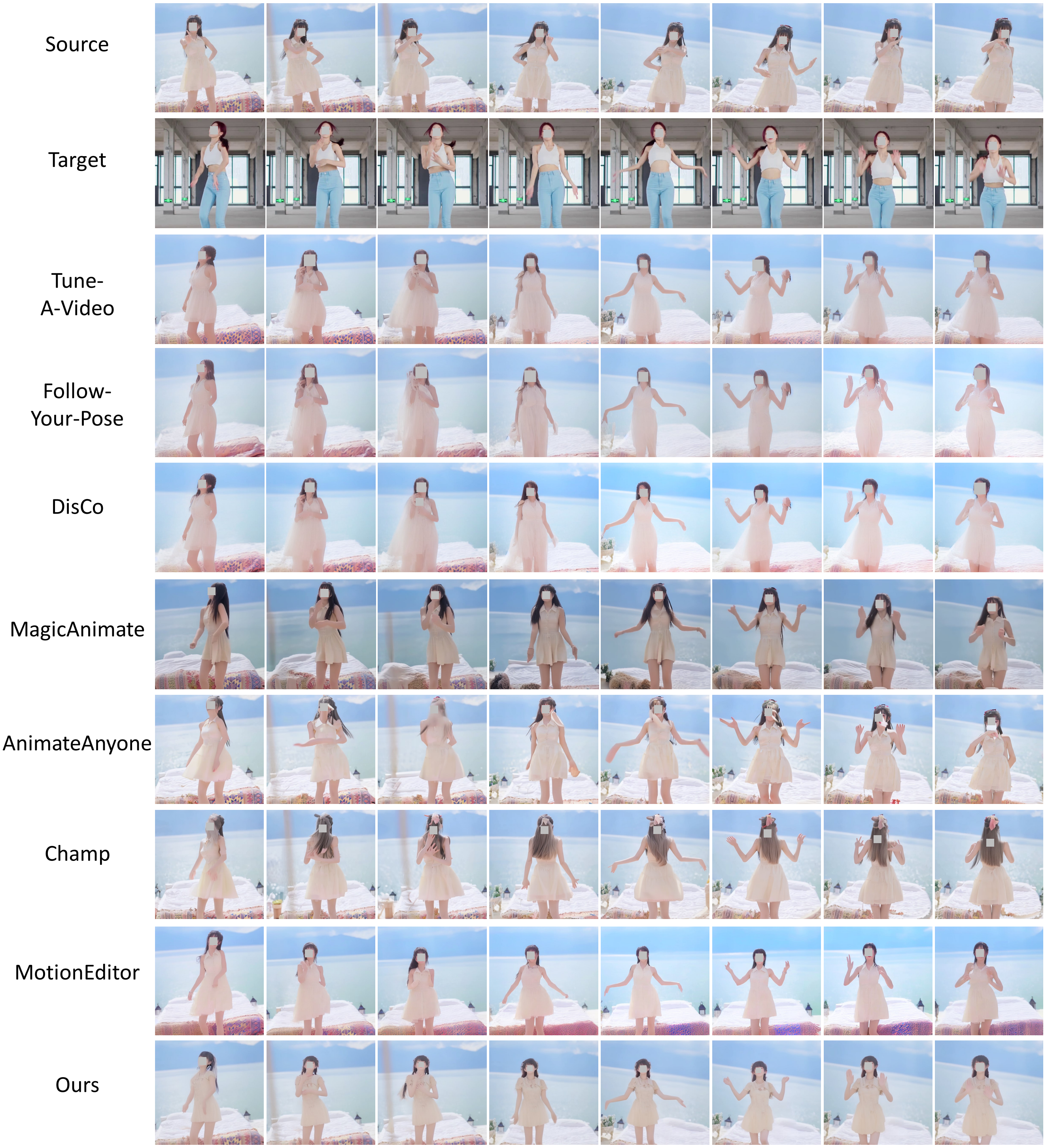

Figure 11. Performance comparison on a video containing large camera movements.

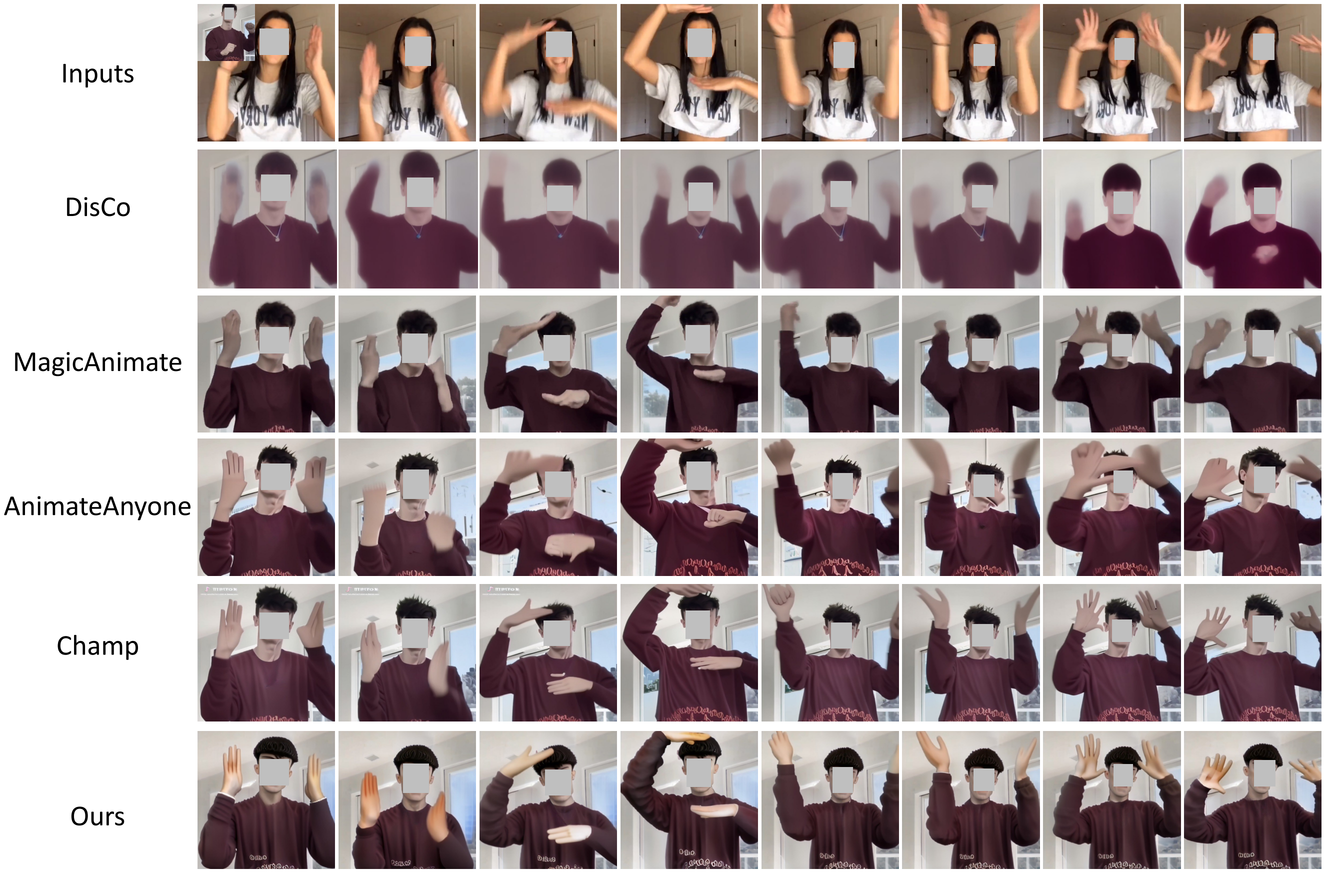

Figure 12. Comparisons between our MotionFollower and other state-of-the-art models regarding the human motion transfer task.

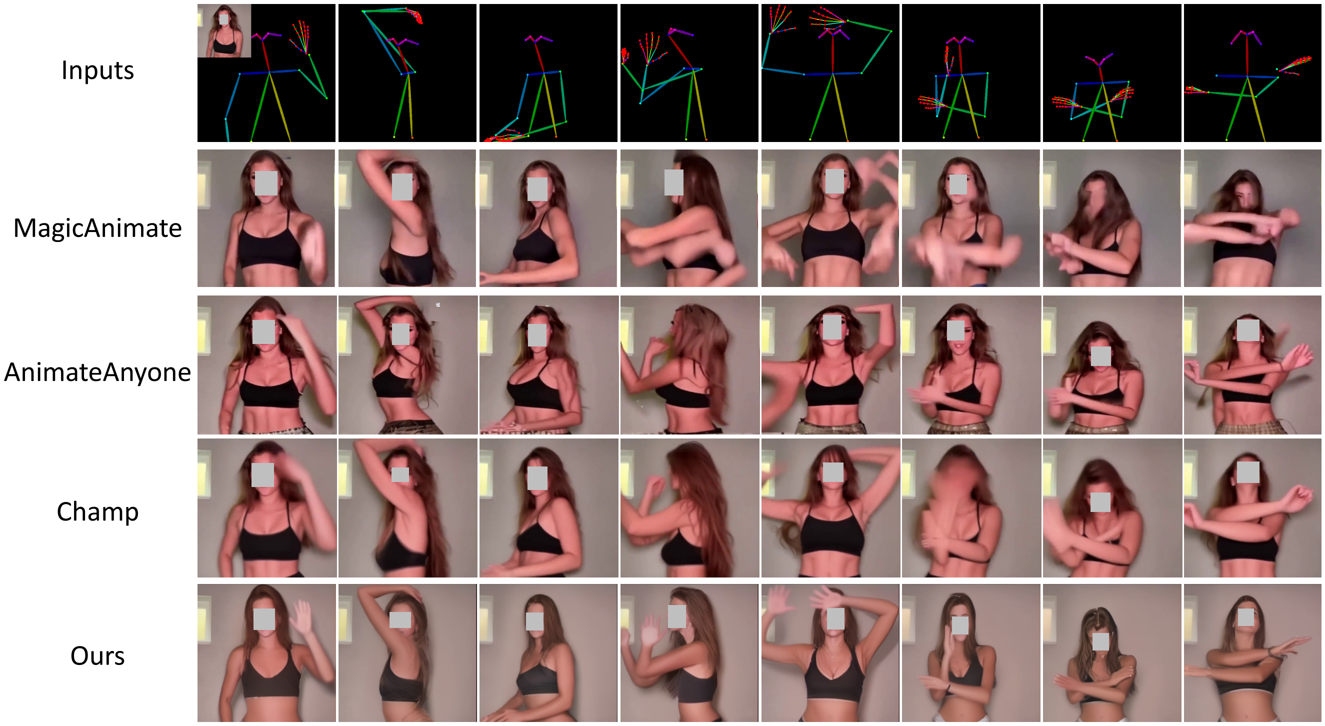

Figure 13. Comparisons between our model and other models regarding human motion transfer, using the same case as in Champ [14].

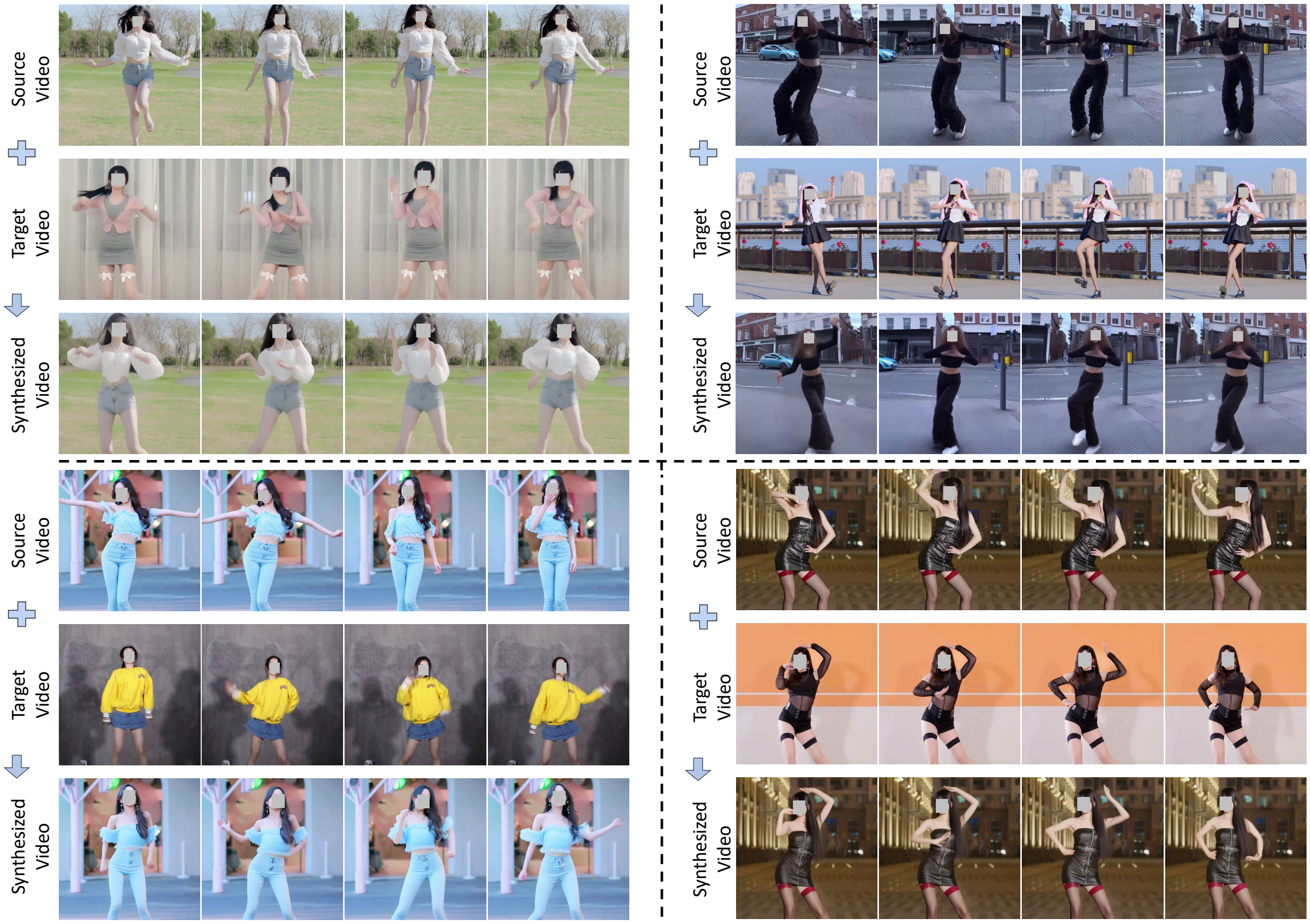

Figure 14. Video motion editing results of our MotionFollower (1/5).

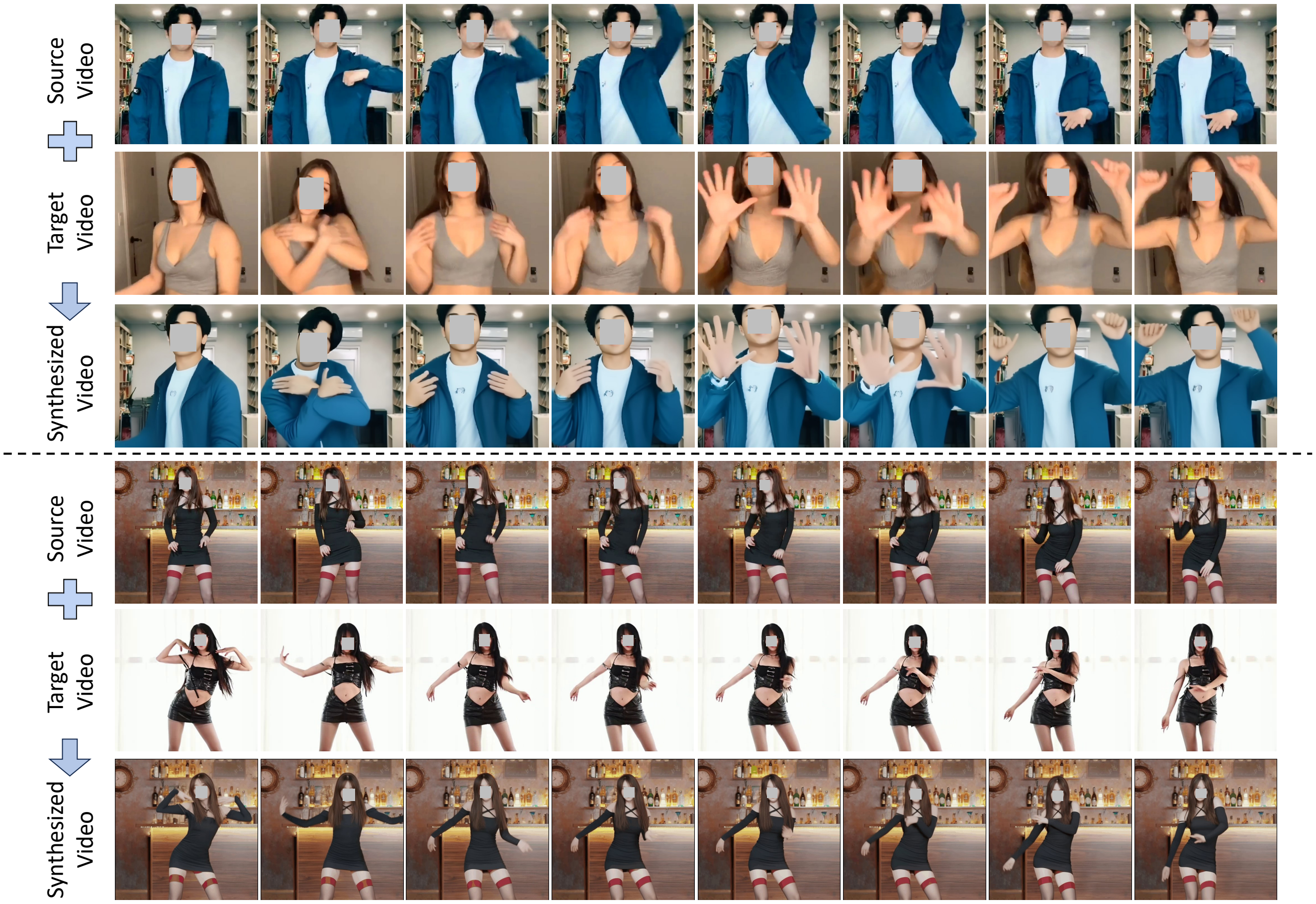

Figure 15. Video motion editing results of our MotionFollower (2/5).

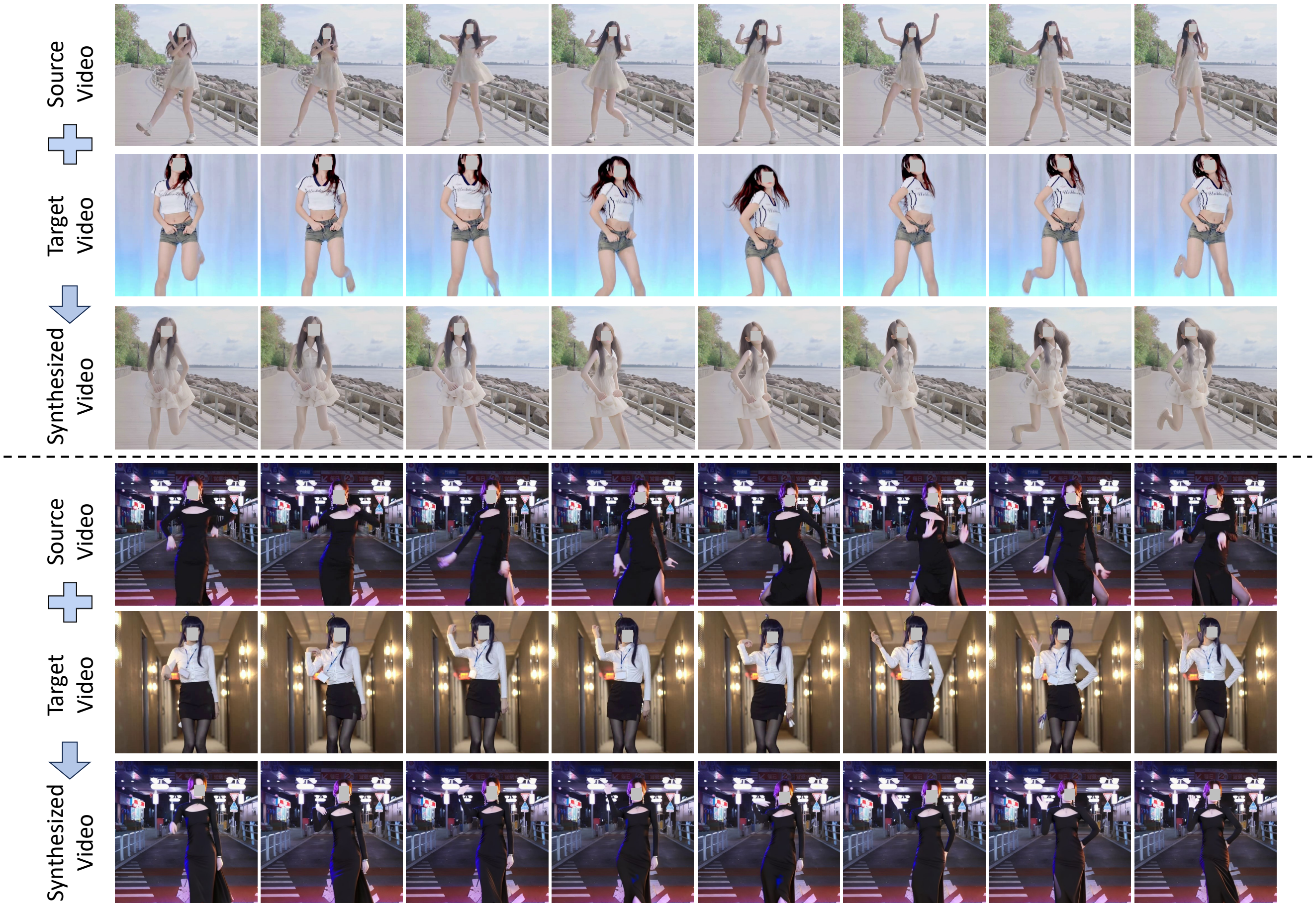

Figure 16. Video motion editing results of our MotionFollower (3/5).

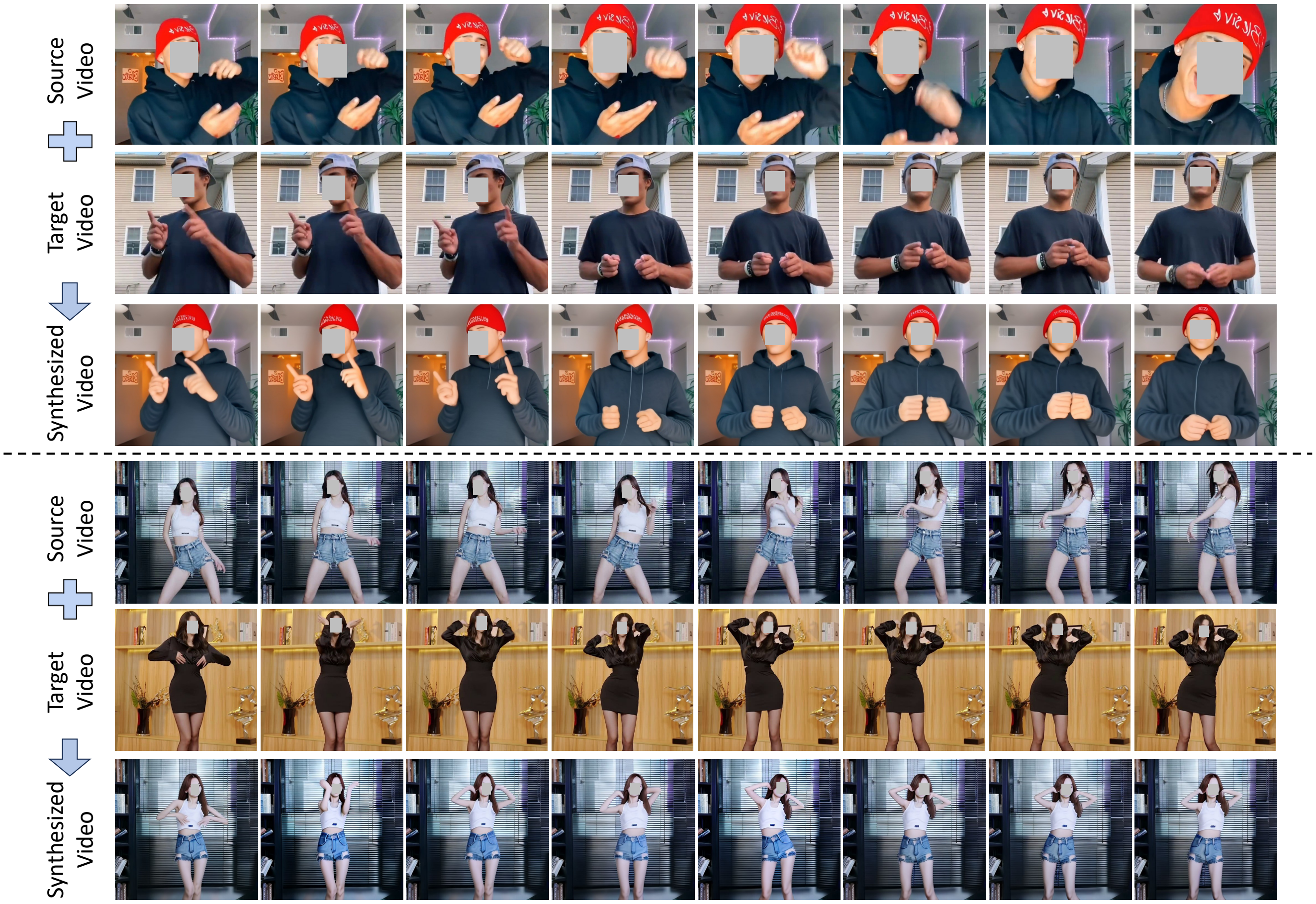

Figure 17. Video motion editing results of our MotionFollower (4/5).

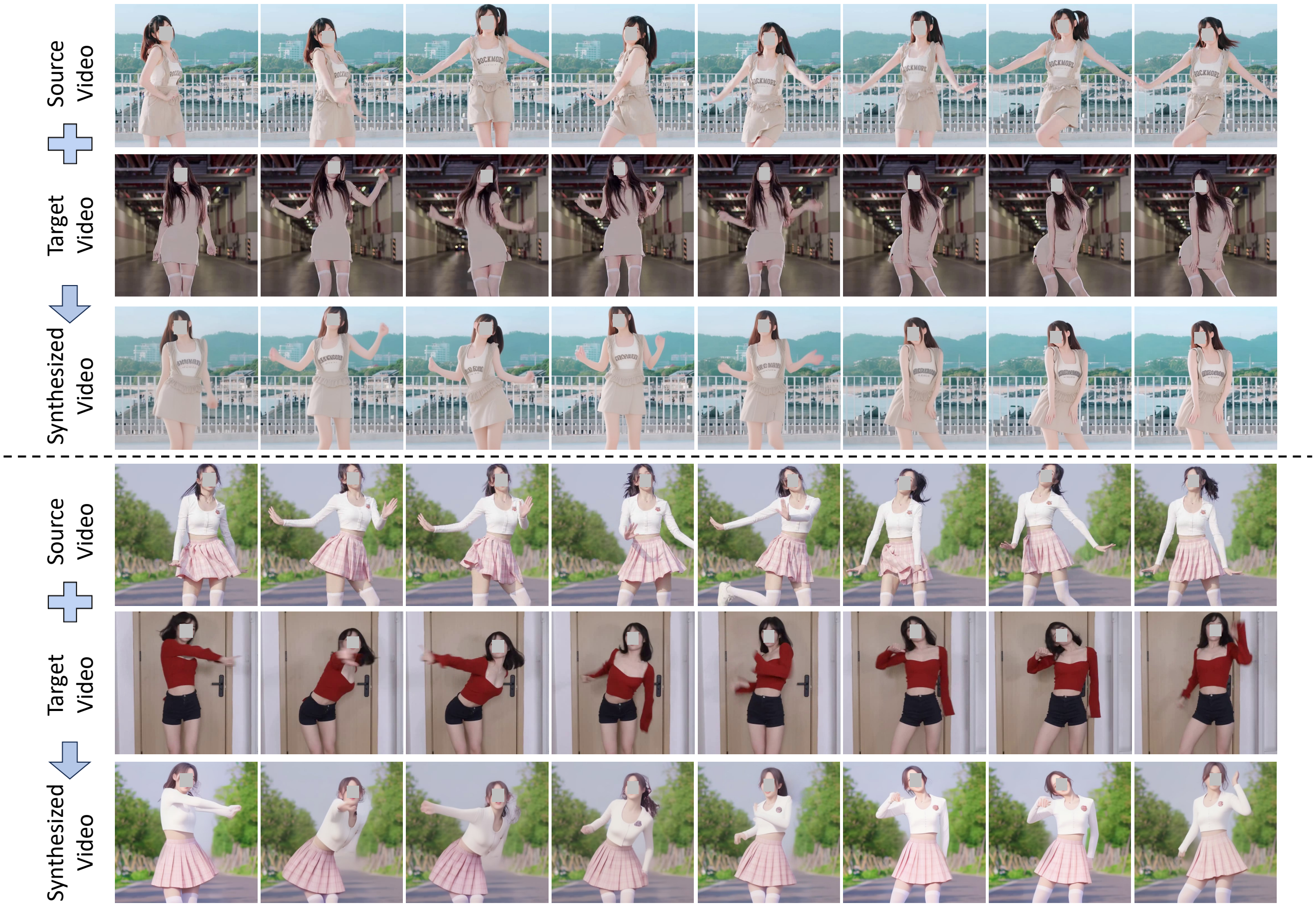

Figure 18. Video motion editing results of our MotionFollower (5/5).

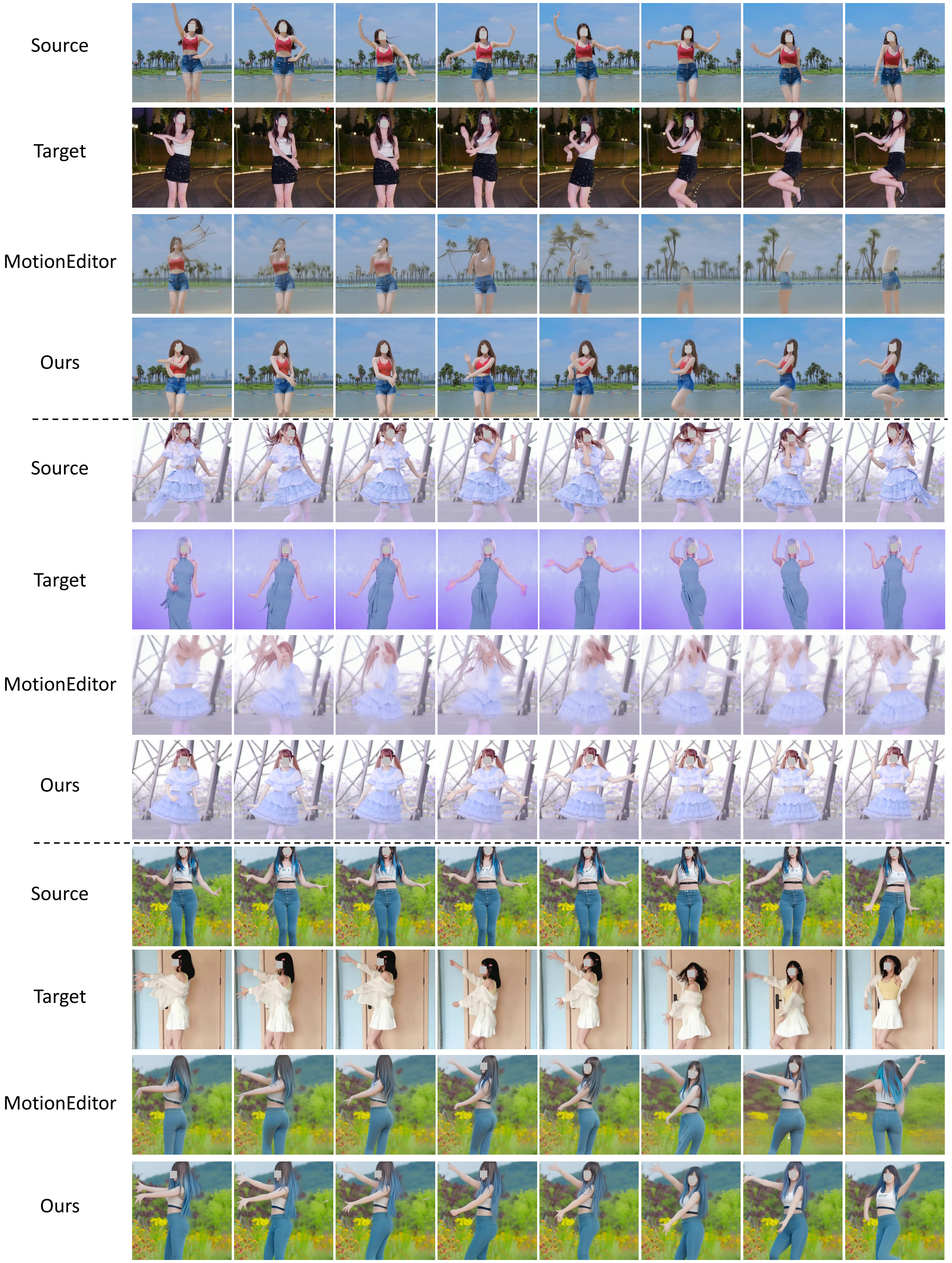

Figure 19. Additional comparison results between our MotionFollower and the strongest competitor MotionEditor.

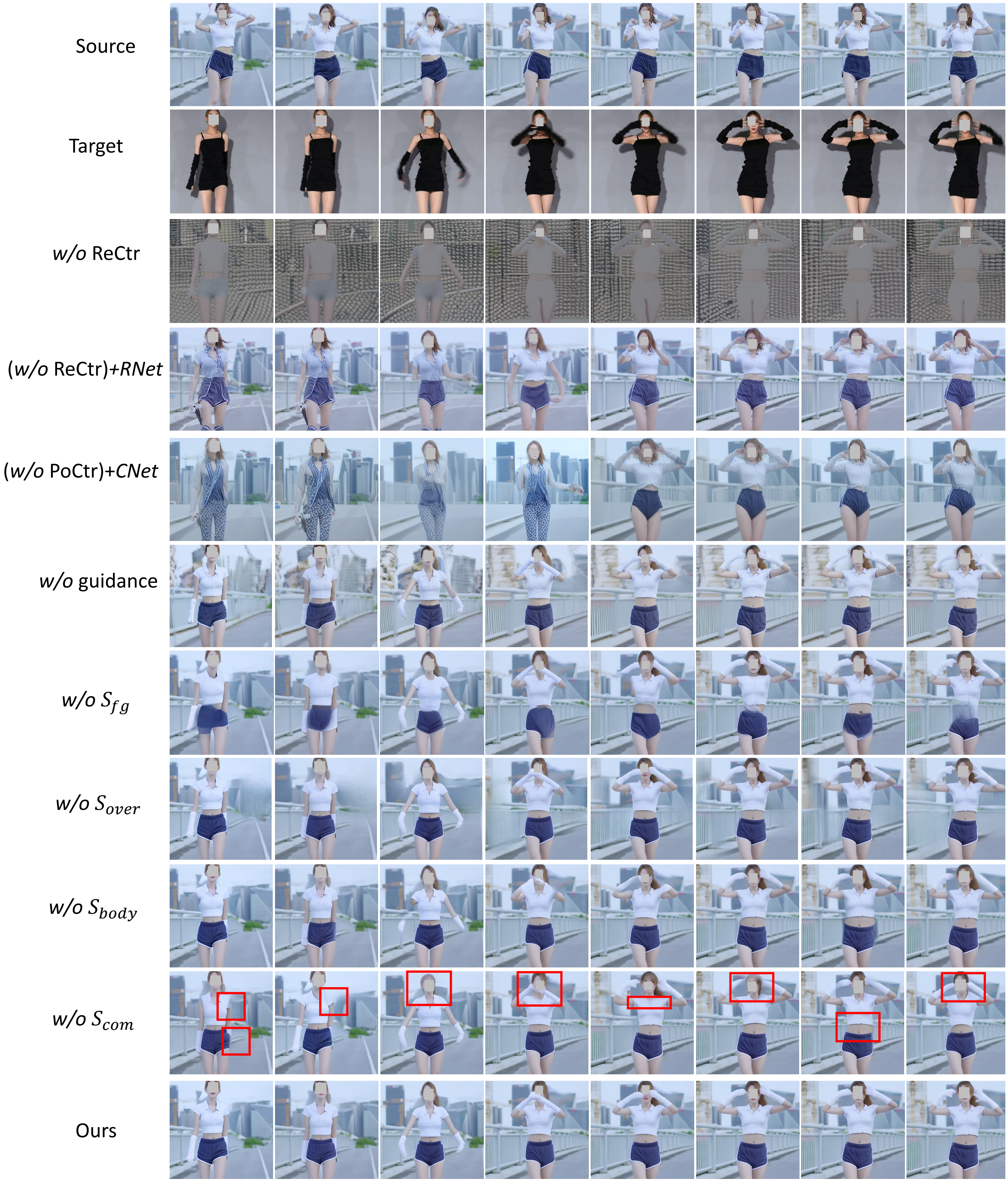

Figure 20. A more comprehensive ablation study result.

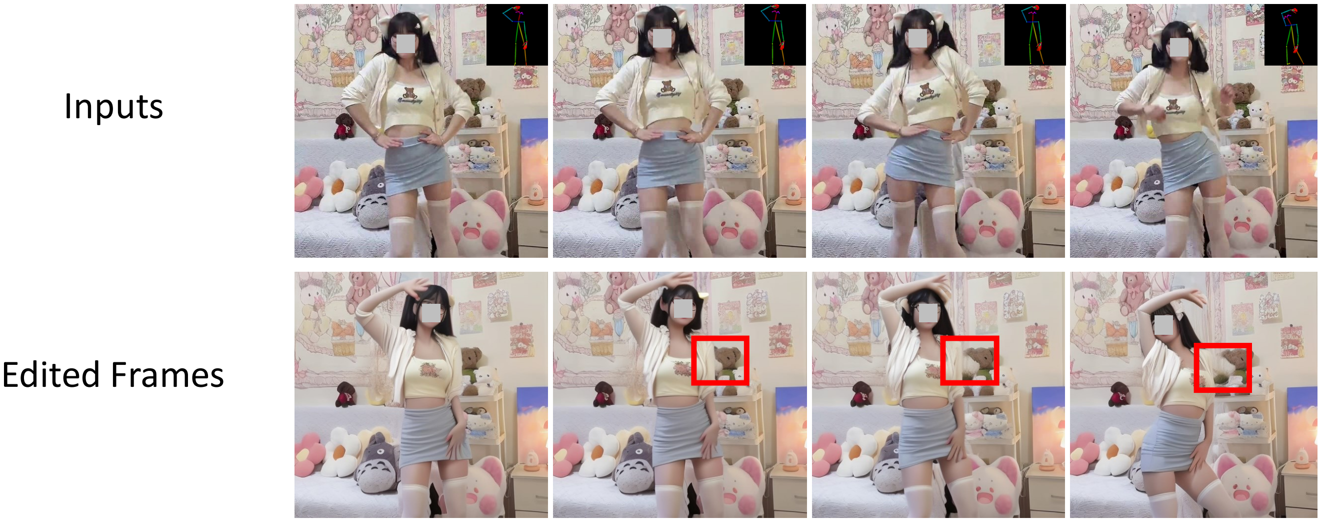

Figure 21. One failure case of our MotionFollower.