

Supplementary Materials for

“MIMO: Controllable Character Video Synthesis with Spatial Decomposed Modeling”

Yifang Men, Yuan Yao, Miaomiao Cui, Liefeng Bo

Tongyi Lab, Alibaba Group

In this document we provide the following supplementary contents:

- Video demonstration of MIMO.
- Details of structured motion code.
- Details of network architectures.
- Target body shape adaptation.
- Long video generation.
- Results of controllable character video synthesis.
- Comparisons with SOTA methods.
- Limitations and future work.

1. Video demonstration of MIMO

We provide a video demonstration (file '6054_video.mp4') for MIMO, showcasing the task definition, motivation, and applications, as well as additional comparison results.

Task. MIMO is a novel framework for synthesizing realistic character videos with controllable attributes (i.e., character, motion and scene) provided by simple user inputs. Given a single reference image of character, it can generate animated avatars in driving 3D poses retrieved from motion datasets or extracted from in-the-wild videos. Real-world scenes from driving videos can also be integrated into the synthesis with natural human-object interactions, thus enabling a brand-new task of video character replacement.

Motivation. Previous 3D works typically require multi-view captures for per-case training, and 2D methods only allows for character animation in simple motions. In contrast, we present MIMO as a pretrained, generalizable model to simultaneously achieve advanced scalability to arbitrary characters, generality to novel 3D motions, and applicability to interactive real-world scenes in a unified framework.

Applications. We present controllable results of our method with arbitrary characters, novel 3D motions and interactive scenes. It covers diverse applications including original character animation and the new task of video character replacement.

Comparisons. We provide more video results for visual comparison with state-of-the-art methods: Animate Anyone [1], Mimic-Motion [2] and Champ [3].

2. Details of structured motion code.

As shown in Figure 1, we define a set of latent codes $\mathcal{Z} = \{z_1, z_2, \dots, z_{6890}\}$, $z_i \in \mathbb{R}^{1 \times \gamma}$, and anchor them to corresponding vertices of a deformable human body model (SMPL) to obtain the structured body codes. For each frame t , SMPL parameters \mathcal{S}_t and camera parameters \mathcal{C}_t are estimated from the human frame via a 3D human pose parser [4]. Then the spatial locations of the body codes are transformed to posed ones based on the \mathcal{S}_t . Using the differentiable rasterizer [5] with vertex interpolation, the body codes can be projected to the 2D plane based on the camera setting \mathcal{C}_t , thus obtaining the pose feature map $\mathcal{F}_t \in \mathbb{R}^{H \times W \times \gamma}$. The sequences $\{\mathcal{F}_t, t = 1, \dots, N\}$ will be stacked along the time axis and fed into the pose encoder \mathcal{E}_p to obtain the motion code \mathcal{C}_{mo} .

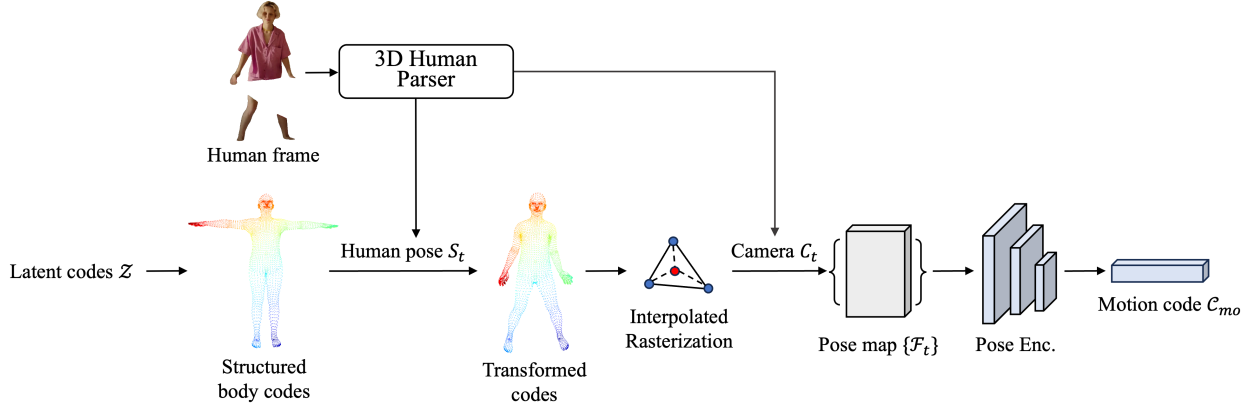


Figure 1: The visualized flowchart of structured motion code.

3. Details of network architectures.

3.1. Pose, identity and scene encoders

The pose encoder consists of 3D convolution and SiLU in each layer to efficiently process temporal feature maps, whose architecture is described in Table 1.

Table 1: Architecture of the pose encoder. Each layer consists of 3D convolution and SiLU.

	Layer Description	Output Dim.
	Input volume	$N \times \gamma \times T \times H \times W$
1	Conv3D + SiLU (3×3 kernel, 16 features, stride 1)	$N \times 16 \times T \times H \times W$
2	Conv3D + SiLU (3×3 kernel, 16 features, stride 1)	$N \times 16 \times T \times H \times W$
3	Conv3D + SiLU (3×3 kernel, 32 features, stride 2)	$N \times 32 \times T \times H/2 \times W/2$
4	Conv3D + SiLU (3×3 kernel, 32 features, stride 1)	$N \times 32 \times T \times H/2 \times W/2$
5	Conv3D + SiLU (3×3 kernel, 96 features, stride 2)	$N \times 96 \times T \times H/4 \times W/4$
6	Conv3D + SiLU (3×3 kernel, 96 features, stride 1)	$N \times 96 \times T \times H/4 \times W/4$
7	Conv3D + SiLU (3×3 kernel, 256 features, stride 2)	$N \times 256 \times T \times H/8 \times W/8$
8	Conv3D + SiLU (3×3 kernel, 320 features, stride 1)	$N \times 320 \times T \times H/8 \times W/8$

For identity encoding, inspired by [1,3], we employ a CLIP image encoder and a reference-net architecture to embed for the global and local feature, respectively. The detailed architectures can be found in their released implements.

For the scene and occlusion encoding, we use a shared and frozen VAE encoder to faithfully extract their latent codes. The detailed architecture of VAE encoder and parameter weights can be found in pretrained SD 1.5 [6].

3.2. Conditional diffusion-based decoder

We adopt a denoising U-Net backbone built upon Stable Diffusion (SD) [6] with temporal layers from [7] to simulate the denoising process in the latent space, and a VAE decoder [8] to convert the denoised result into the video clip. The details of conditional code insertion are shown in Table 2. The full scene code is concatenated with the latent noise, and is fed into a 3D convolution layer for fusion and alignment. The motion code is added to the fused feature and input to the denoising U-Net. For identity code, its local feature and global feature are inserted into the U-Net via self-attention layers and cross-attention layers, respectively, following the implement in [1].

Table 2: Architecture of the conditional diffusion-based decoder.

	Operation	Output Dim.
1	Input: Full scene code	$N \times 8 \times T \times H/8 \times W/8$
2	Input: Noise	$N \times 4 \times T \times H/8 \times W/8$
3	Feature concatenation	$N \times 12 \times T \times H/8 \times W/8$
4	Conv3D (3×3 kernel, 320 features, stride 1)	$N \times 320 \times T \times H/8 \times W/8$
5	Input: Motion code	$N \times 320 \times T \times H/8 \times W/8$
6	Feature addition	$N \times 320 \times T \times H/8 \times W/8$
7	Input: Identity code	$N \times 768$ (clip); stacked U-Net block output (refer-net)
8	Denoising U-Net blocks	$N \times 4 \times T \times H/8 \times W/8$
9	VAE decoding	$N \times 3 \times T \times H \times W$

4. Target body shape adaptation

With the help of the parametric body model SMPL equipped with decoupled pose and shape parameters, our method can easily adjust the motion representation to adapt target characters with various body shapes. For extreme shapes, the shape parameter β can be extracted from the reference image via existing human reconstruction method, and is combined with pose parameters \mathcal{S}_t to compute transformed vertices positions in Figure 1, thus obtaining adapted motion code aligning to the target character. In this way, even extremely different body shapes can be faithfully preserved during the synthesis process. Animation results for characters with significantly different shapes are shown in Figure 2.



Figure 2: Results of diverse characters with significantly different shapes.

5. Long video generation

With 24-frame video clip for training, our model generates long videos with arbitrary length in an inter-clip fusion manner. We use an overlay of 4-frame for clip inference and fusion. Specifically, when inferring a video clip, the first 4 frames of current clip are succeeded from the last 4 frames of the previous clip, and combined with subsequent 20 frames as the input. For each denoising step, the intermediate results of the last 4 frames from the previous clip are overlayed onto the current frames, thus allowing for smooth transitions. Thanks to spatial decomposed modeling and effective controls in appearance and motion, our model can synthesize faithful character with stable scene across multiple clips to some extent. But it is worthy of noting that the long-term temporal consistency is not the primary focus of our work and still requires further exploration for better results.

6. Results of controllable character video synthesis

With the controllable character video synthesis framework in advanced capabilities for arbitrary characters, novel 3D motions and interactive scenes, MIMO enables not only more realistic results for original character animation, but also a brand-new task of video character replacement. The frame results are shown in Figure 3 and Figure 4.



Figure 3: Results of character animation with arbitrary characters and novel 3D motions.

Driving video with “complex 3D motion” and “interactive scene”

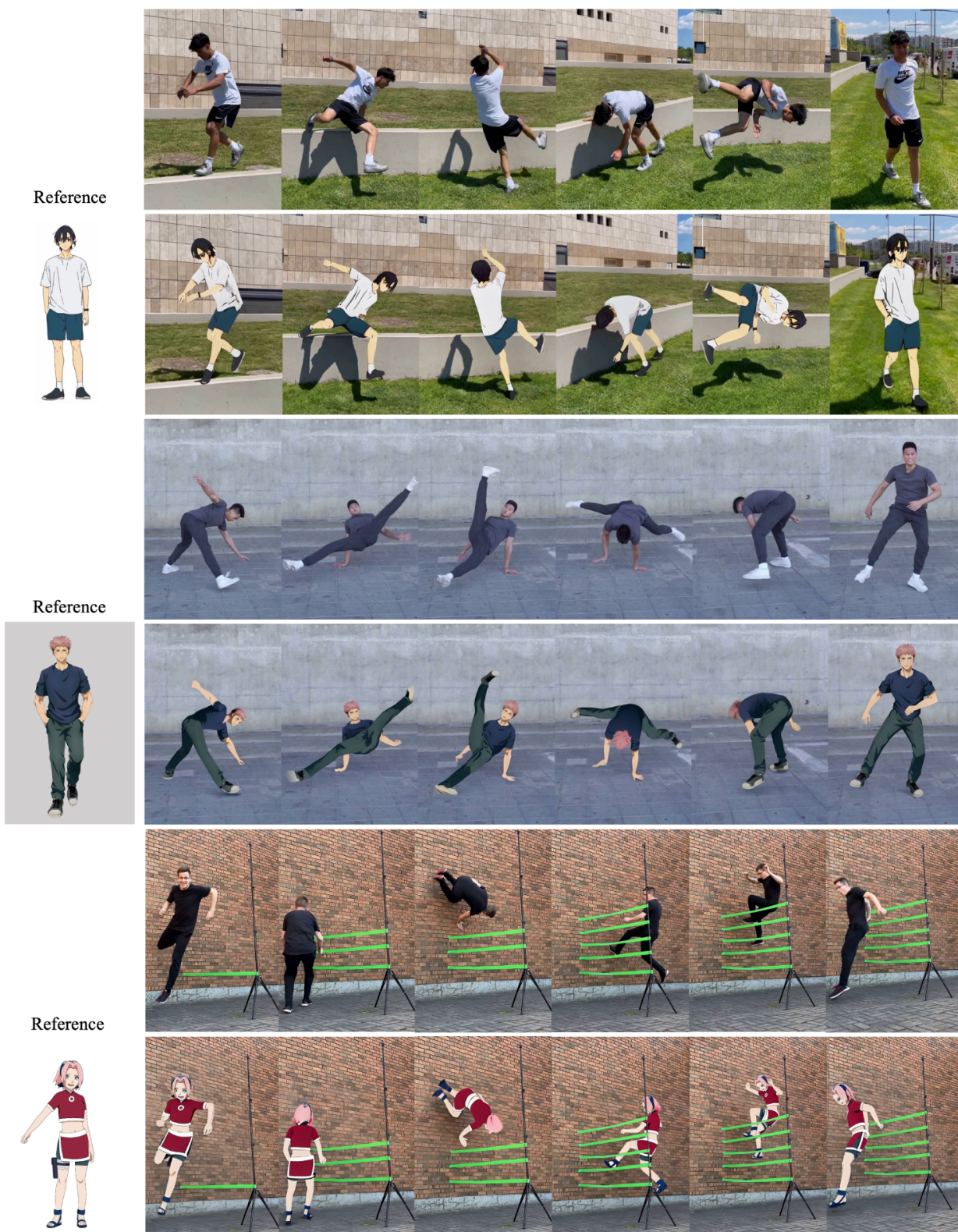


Figure 4: Results of video character replacement by extracting motion and scene attributes from the driving video.

7. Comparisons with state-of-the-art methods

More frame results for clearer comparison are shown in Figure 5.

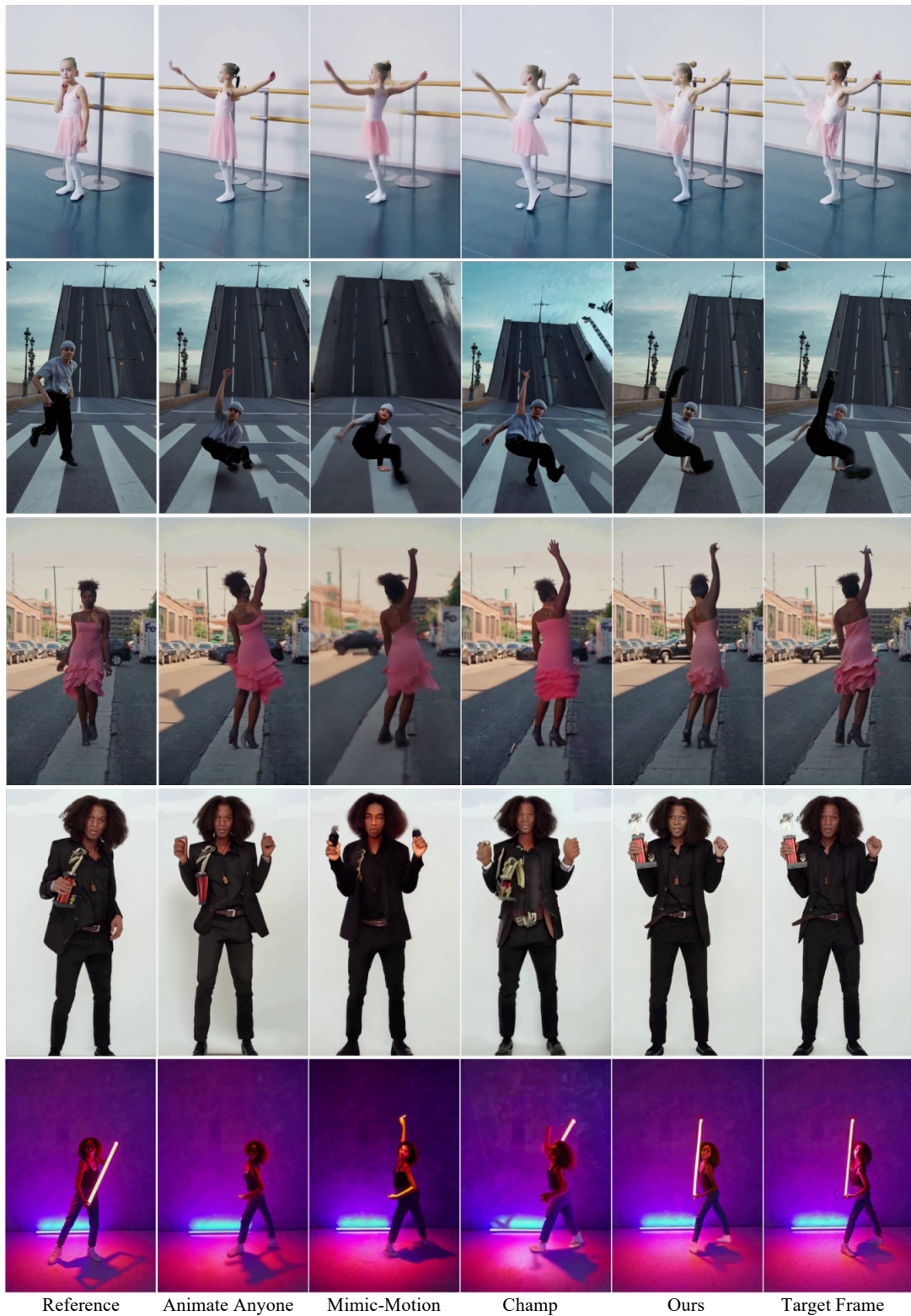


Figure 5: Comparison with state-of-the-art methods: Animate Anyone [1], Mimic-Motion [2] and Champ [3].

Considering insufficient scene modeling of previous methods, we also provide additional quantitative comparison by removing background and object for only character synthesis in Table 3.

Table 3: Additional quantitative comparison for only character synthesis.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	FVD \downarrow
Animate Anyone	22.604	0.763 \uparrow	0.182	184.7
Mimic-Motion	22.836	0.752	0.224	162.8
Champ	23.163	0.814	0.237	221.5
Ours	25.891	0.897	0.121	125.2

8. Limitations and future work

Our method offers easy user control by accepting a single image for character reference. While our model can generate realistic 360° views with inter-frame consistency, it struggles to maintain consistent complex texture patterns on the human back over long-term intervals, particularly during re-appearances after disappearing (see Figure 6 (a, b)). This issue arises from limited long-term memory and single-view input with only frontal observation. Actually, our framework could be easily adapted to accommodate multi-view inputs for appearance encoding, potentially resolving the issue by providing multi-view references such as frontal and posterior images. Alternatively, exploring long-term memory for generative models could also address this problem and holds significant research value for further works.

We applied SMPL, a deformable human body model, for 3D motion representation in character video synthesis. Extending the unified motion representations (e.g., 3D tracking points, 3D flow for objects) could open up new possibilities for common object animation and editing in other applications.



Figure 6: The failure case of inconsistent complex texture patterns on the human back over re-appearances after disappearing.

Reference

- [1] Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8153–8163, 2024.
- [2] Yuang Zhang, Jiayi Gu, Li-Wen Wang, Han Wang, Junqi Cheng, Yuefeng Zhu, and Fangyuan Zou. Mimicmotion: High-quality human motion video generation with confidence-aware pose guidance. arXiv preprint arXiv:2406.19680, 2024.
- [3] 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. arXiv preprint arXiv:2403.14781, 2024.
- [4] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 14783–14794, 2023.
- [5] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. ACM Transactions on Graphics, 39(6), 2020.
- [6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- [7] 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.
- [8] Diederik P Kingma. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.