

# MoCha: End-to-End Video Character Replacement without Structural Guidance

## Supplementary Material

### 1. Appendix

In the supplementary material, we provide the following contents:

- Additional implementation details of our dataset construction;
- More analysis and qualitative results of MoCha;
- A 2-minute video comparing our stabilized videos with results by other methods;

#### A. Details of Data Construction

In this section, we provide more details of how the three datasets are constructed to train MoCha. Some data examples are shown in Fig. 1.

##### A.1. Rendered Data

We construct the following sets to render our dataset:

**Scene.** We collect 50 different 3D environments from the UE ecosystem. For each scene, we mark 10 to 25 points suitable to place the character. Each point is recorded via a location vector and a rotation vector. We filter some scenes which are too small to mark sufficient points.

**Characters.** We collect 110 characters from the UE ecosystem and Mixamo website [6] with different styles. Furthermore, we collect 66 characters from MetaHuman [7], which support facial expression controls.

**Animations.** We collect 200 animation clips from the UE ecosystem and the Mixamo website [6] and retarget them to the different character models. We filter some animations that exhibit excessively large or chaotic movements.

**Expressions.** We collect 1000 facial expression sequences from the UE ecosystem, including different emotions such as anger, surprise and fear.

**Camera Trajectory.** We incorporate a specific camera trajectory into every video sequence. Concretely, we use the character’s head position as the center of a sphere and randomly sample the camera’s initial position within a predefined radial range. We ensure that the sampled point lies on the hemisphere facing the character and is constrained vertically to prevent positions that are excessively high or low. Finally, we orient the camera so that it focuses on the character’s head in the first frame.

Following Recamaster [1], we apply five camera trajectory generation rules, and uniformly sample from this set to determine the trajectory for each video:

- **Pan and Tilt:** The camera’s pan angle or tilt angle is randomly rotated. The camera’s location remains fixed.
- **Basic Translation:** The camera is translated along one of the  $x$ ,  $y$ , or  $z$  axes. The camera’s orientation remains fixed.
- **Basic Arch:** The camera rotates by a certain angle around the character’s vertical ( $z$ ) axis.
- **Random Trajectory:** The camera follows a trajectory defined by a sequence of 1 to 3 randomly selected key points.
- **Static Camera:** The camera’s position and orientation remain fixed.

In total, we synthesize approximately 50K single-character rendered videos and 20K multi-character rendered videos.

##### A.2. Expression-Driven Face Animation Data

We collect 20K movie shots containing characters for use in our video inpainting process. We then utilize the same facial expression set (detailed in Sec. A.1) to drive these images. In total, we construct approximately 20K paired data using this strategy.

##### A.3. Augmented Video-Mask Data

We initially collect approximately 100K raw samples each from the VIVID [4] and VPData [2] datasets. We then utilize a YOLO-v12 [8] detector to filter this collection, retaining only those clips that contain a detectable human face and where the face occupies more than 10% screen area. Ultimately, we retain a total of 120K data samples and use 10K for training.

Next, we element-wise multiply the mask with the original video to “paint out” the character while preserving the background, producing an edited clip in which the subject is grayed and the environment remains intact. We use this clip as the source video and the original video as the target, and randomly sample one frame from the target as the reference image to condition identity.

### B. More Ablation Studies

In this section, we provide additional qualitative results generated by MoCha. These examples further demonstrate the capabilities of our framework.

#### B.1. Ablation on Number of References

MoCha supports arbitrarily many reference images as input to condition the target character’s identity. We conduct a comparison to evaluate the influence of varying the number

of reference images on the generation quality. The results are shown in Fig. 2.

When only a single full-body reference is provided, the character’s face occupies only a small region of the image. As a result, the model lacks sufficient facial information to accurately preserve identity. Consequently, providing an additional facial reference efficiently improves performance. However, the results also demonstrate that this gap can be effectively mitigated by our post-training strategy (an Identity-Enhancing LoRA). By integrating the LoRA module, videos generated using only a single reference image achieve a comparably high level of identity preservation.

### B.2. Ablation on the Choice of Designated Mask

We evaluate our model’s sensitivity to the frame index in Fig. 3. The results show that when masks from different frames are used as input, MoCha consistently generates high-quality and well-aligned videos, despite exhibiting only minor variations in the output.

### B.3. Ablation on Identity-Enhancing Post-Training

Post-training significantly boosts face cosine similarity from 68.22% to 87.55%. Meanwhile, it maintains or slightly improves other quality metrics on VBench[5], including temporal flickering (98.45 vs. 98.48), aesthetic quality (57.00 vs. 57.48), and motion smoothness (99.05 vs. 99.05). This shows our post-training effectively enhances identity fidelity while preserving overall video quality.

### B.4. Ablation on the Reward Setting

We conduct an ablation experiment to evaluate the necessity of pixelwise reward. As shown in Fig. 4, using identity reward alone leads to **reward hacking**, where the model tends to generate multiple redundant faces. Thus, the pixelwise reward is essential to maintain the base generation quality.

### B.5. Ablation on the Post-Training Iteration

ArcFace[3] can interfere with expression and head pose. We mitigate this by carefully controlling the optimization iterations during post-training. Results with insufficient and excessive training are shown in Fig. 5.

## C. More Results

### C.1. More Comparison with SOTA Methods

Fig. 8 shows more qualitative comparison with the state-of-the-art methods. The results further illustrate MoCha’s capacity to preserve animation fidelity and multi-character interaction dynamics.

### C.2. MoCha beyond Character Replacement

Fig. 6 showcases MoCha’s broad application beyond the core character replacement task. It demonstrates that

MoCha supports a wide range of subject replacement, extending far beyond human characters. Furthermore, by editing the reference image through techniques like image inpainting or by applying prompt engineering methods to guide the identity, MoCha enables precise local editing on parts of the character, such as changing the face or clothing.

### C.3. Failure Cases

Since we rely on the tracking capabilities of the video generation model to follow the character, MoCha struggles with certain shortcomings inherited from it. For instance, MoCha may generate artifacts when the source video contains motion blur resulting from high-speed movement. Additionally, when characters cross paths in the video, the model occasionally tracks the incorrect subject. These specific failure cases are presented in Fig. 7.

## References

- [1] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuoqiu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. *arXiv preprint arXiv:2503.11647*, 2025. 1
- [2] Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. Videopainter: Any-length video inpainting and editing with plug-and-play context control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. 1
- [3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 2
- [4] Jiahao Hu, Tianxiong Zhong, Xuebo Wang, Boyuan Jiang, Xingye Tian, Fei Yang, Pengfei Wan, and Di Zhang. Vivid-10m: A dataset and baseline for versatile and interactive video local editing. *arXiv preprint arXiv:2411.15260*, 2024. 1
- [5] Ziqi Huang, Yanan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 2
- [6] Maximo. Maximo. <https://www.mixamo.com/>, 2025. 1
- [7] MetaHuman. Metahuman. <https://www.metahuman.com/>, 2025. 1
- [8] Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025. 1



Figure 1. Examples of three constructed dataset.



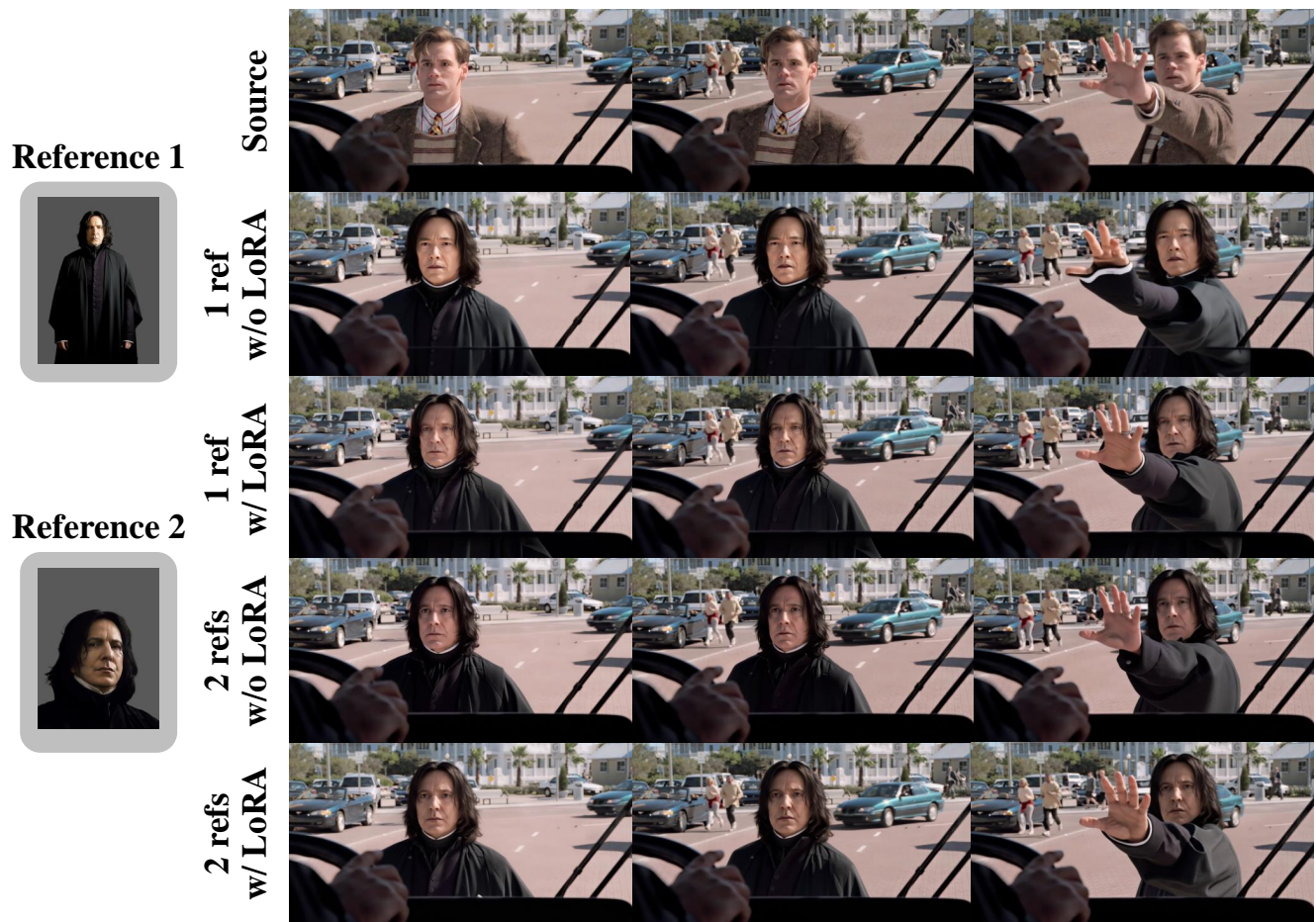


Figure 2. Ablation on Number of References.



Figure 3. Ablation on the Choice of Mask.



Figure 4. Ablation on the Reward Setting.

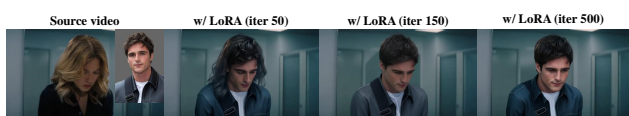


Figure 5. Ablation on the Post-Training Iteration.

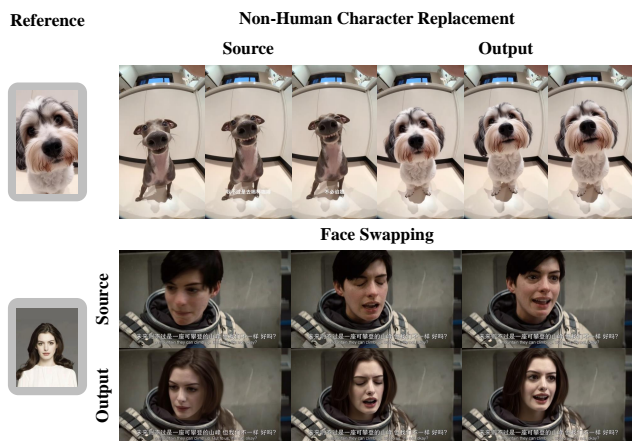


Figure 6. Other Application of MoCha.

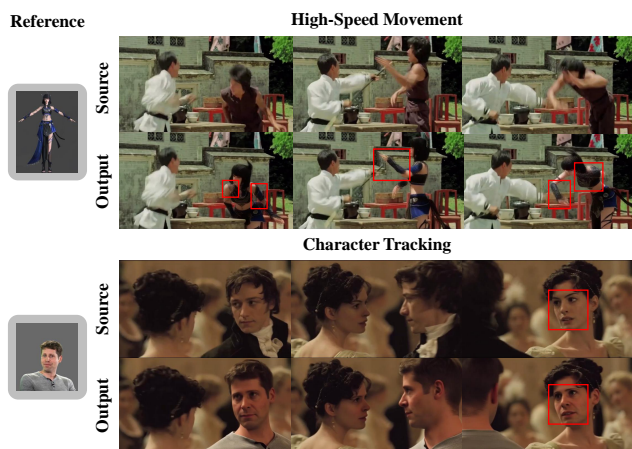


Figure 7. Failure Cases of MoCha.





Figure 8. More comparison with state-of-art methods.