ReCamMaster: Camera-Controlled Generative Rendering from A Single Video

Supplementary Material

A. Introduction of the Base Text-to-Video Generation Model

We use a transformer-based latent diffusion model [37] as the base T2V generation model, as illustrated in Fig. 7. We employ a 3D-VAE to transform videos from the pixel space to a latent space, upon which we construct a transformer-based video diffusion model. Unlike previous models that rely on UNets or transformers, which typically incorporate an additional 1D temporal attention module for video generation, such spatially-temporally separated designs do not yield optimal results. We replace the 1D temporal attention with 3D self-attention, enabling the model to effectively perceive and process spatiotemporal tokens, thereby achieving a high-quality and coherent video generation model. Specifically, before each attention or feed-forward network (FFN) module, we map the timestep to a scale, thereby applying RMSNorm to the spatiotemporal tokens.

B. Details of Data Construction

In this section, we provide a detailed description of the rendered dataset used to train ReCamMaster.

3D Environments We collect 40 different 3D environments assets from https://www.fab.com/. To minimize the domain gap between rendered data and real-world videos, we primarily select visually realistic 3D scenes, while choosing a few stylized or surreal 3D scenes as a supplement. To ensure data diversity, the selected scenes cover a variety of indoor and outdoor settings, such as city streets, shopping malls, cafes, office rooms, and the countryside.

Characters We collected 70 different human 3D models as characters from https://www.fab.com/ and https://www.mixamo.com/#/, including realistic, anime, and game-style characters.

Animations We collected approximately 100 different animations from https://www.fab.com/ and https://www.mixamo.com/#/, including common actions such as waving, dancing, and cheering. We used these animations to drive the collected characters and created diverse datasets through various combinations.

Camera Trajectories Due to the wide variety of camera movements, amplitudes, shooting angles, and camera parameters in real-world videos, we need to create as diverse camera trajectories and parameters as possible to cover various situations. To achieve this, we designed some rules to batch-generate random camera starting positions and movement trajectories:

1. Camera Starting Position.

We take the character's position as the center of a hemisphere with a radius of 10m and randomly sample within this range as the camera's starting point, ensuring the closest distance to the character is greater than 0.5m and the pitch angle is within 45 degrees.

2. Camera Trajectories.

- **Pan & Tilt:** The camera rotation angles are randomly selected within the range, with pan angles ranging from 5 to 60 degrees and tilt angles ranging from 5 to 45 degrees, with directions randomly chosen left/right or up/down.
- **Basic Translation:** The camera translates along the positive and negative directions of the xyz axes, with movement distances randomly selected within the range of $[\frac{1}{4}, 1] \times$ distance2character.
- **Basic Arc Trajectory:** The camera moves along an arc, with rotation angles randomly selected within the range of 5 to 60 degrees.
- **Random Trajectories:** 1-3 points are sampled in space, and the camera moves from the initial position through these points as the movement trajectory, with the total movement distance randomly selected within the range of $[\frac{1}{4}, 1] \times$ distance2character. The polyline is smoothed to make the movement more natural.
- **Static Camera:** The camera does not translate or rotate during shooting, maintaining a fixed position.

3. Camera Movement Speed.

To further enhance the richness of trajectories and improve our model's generalization ability, 50% of the training data uses constant-speed camera trajectories, while the other 50% uses variable-speed trajectories generated by nonlinear functions. Consider a camera trajectory with a total of f frames, starting at location L_{start} and ending at position L_{end}. The location at the i-th frame is given by:

$$L_i = L_{start} + (L_{end} - L_{start}) \cdot \left(\frac{1 - \exp(-a \cdot i/f)}{1 - \exp(-a)} \right), \tag{10}$$

where a is an adjustable parameter to control the trajectory speed. When $a > 0$, the trajectory starts fast and then slows down; when $a < 0$, the trajectory starts slow and then speeds up. The larger the absolute value of a, the more drastic the change.

4. Camera Parameters.

We chose two sets of commonly used camera parameters: focal=35mm, aperture=2.8, and focal=24mm, aperture=10.

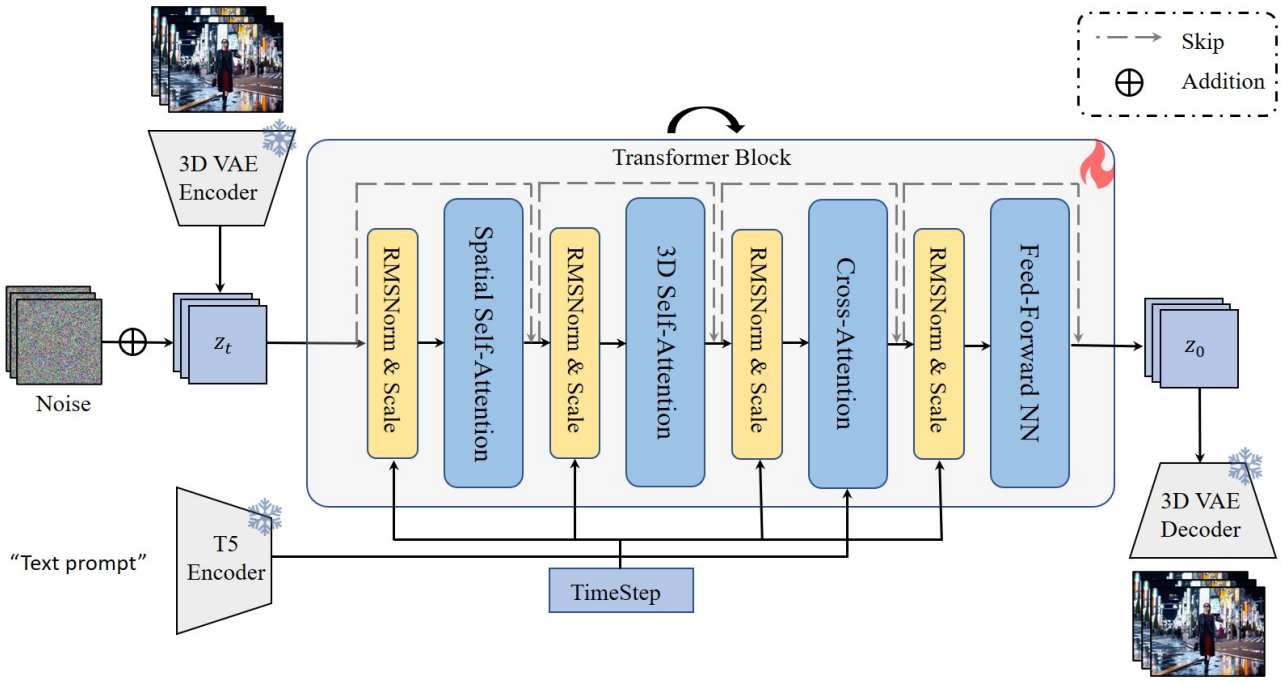

Figure 7. Overview of the base text-to-video generation model.

Figure 8. Rendered multi-camera synchronized dataset.

Table 5. Ablation study on training data construction.

	Visual Quality				Camera Accuracy		View Synchronization		
Dataset	FID ↓	FVD ↓	CLIP-T ↑	CLIP-F ↑	RotErr ↓	TransErr ↓	Mat. Pix.(K) ↑	FVD-V ↓	CLIP-V ↑
Toy Data	69.35	179.22	34.28	**98.77**	1.98	5.24	862.59	89.58	89.70
High-Quality Data	**57.10**	**122.74**	**34.53**	98.74	**1.22**	**4.85**	**906.03**	**90.38**	**90.36**

Figure 9. Unify camera-controlled tasks with ReCamMaster. ReCamMaster supports T2V, I2V, and V2V camera-controlled generation.

C. More Results

C.1. Ablation on Dataset Construction

In our experiments, we find that constructing a diverse training dataset that closely resembles the distribution of real-world videos significantly enhances the model's generalization ability. To demonstrate this, we quantitatively compared the model performance trained on the "toy data" constructed in the early stages of the experiment and the "high-quality data" used in this paper. Specifically, the "toy data" was constructed with 500 scenes in a single 3D environment, and we manually created 20 camera trajectories assigned to 5000 cameras. As a result, this dataset lacked diversity in terms of scenes and camera trajectories, which limited the model's generalization ability on in-the-wild videos and novel camera trajectories. In contrast, the "high-quality data" used in the paper contains 136K videos shot from 13.6K different dynamic scenes in 40 3D environments with 122K different camera trajectories. We present the results in Tab. 5. It is observed that the model shows significant improvements in visual quality, camera accuracy, and synchronization metrics.

C.2. More Results of ReCamMaster

More synthesized results of ReCamMaster are presented in Fig. 12. Please visit our project page for more results. We also showcase the ability of ReCamMaster to support T2V, I2V, V2V camera-controlled tasks in Fig. 9.

C.3. More Comparison with SOTA Methods

Please refer to Fig. 13 for qualitative comparison with the state-of-the-art methods.

Figure 10. Results on non-overlapping first frames.

Figure 11. Visualization of failure cases.

C.4. Results on Non-overlapping First Frames

In the main text, we assume that the first frame of the generated video coincides with the first frame of the input video when testing ReCamMaster. This means the generated video starts from the original video's first frame. To evaluate the method's generalizability, we also rendered 27K 'non-overlapping first-frame' videos as training data and tested whether the generated first frame could differ from the original. Fig. 10 presents the qualitative results, with the leftmost column showing the first frame. It is evident that the model generalizes well to non-overlapping first frames, enabling re-filming from a completely new perspective, thus demonstrating the method's generalizability.

C.5. Failure Cases Visualization

We present the failure cases in Fig. 11. Since our model is built upon a text-to-video base model, we also inherit some of the base model's shortcomings. For instance, the generated hand movements of characters may exhibit inferior quality, as shown in the first and second rows. Moreover, generating very small objects sometimes results in failures, as shown in the third and fourth rows.

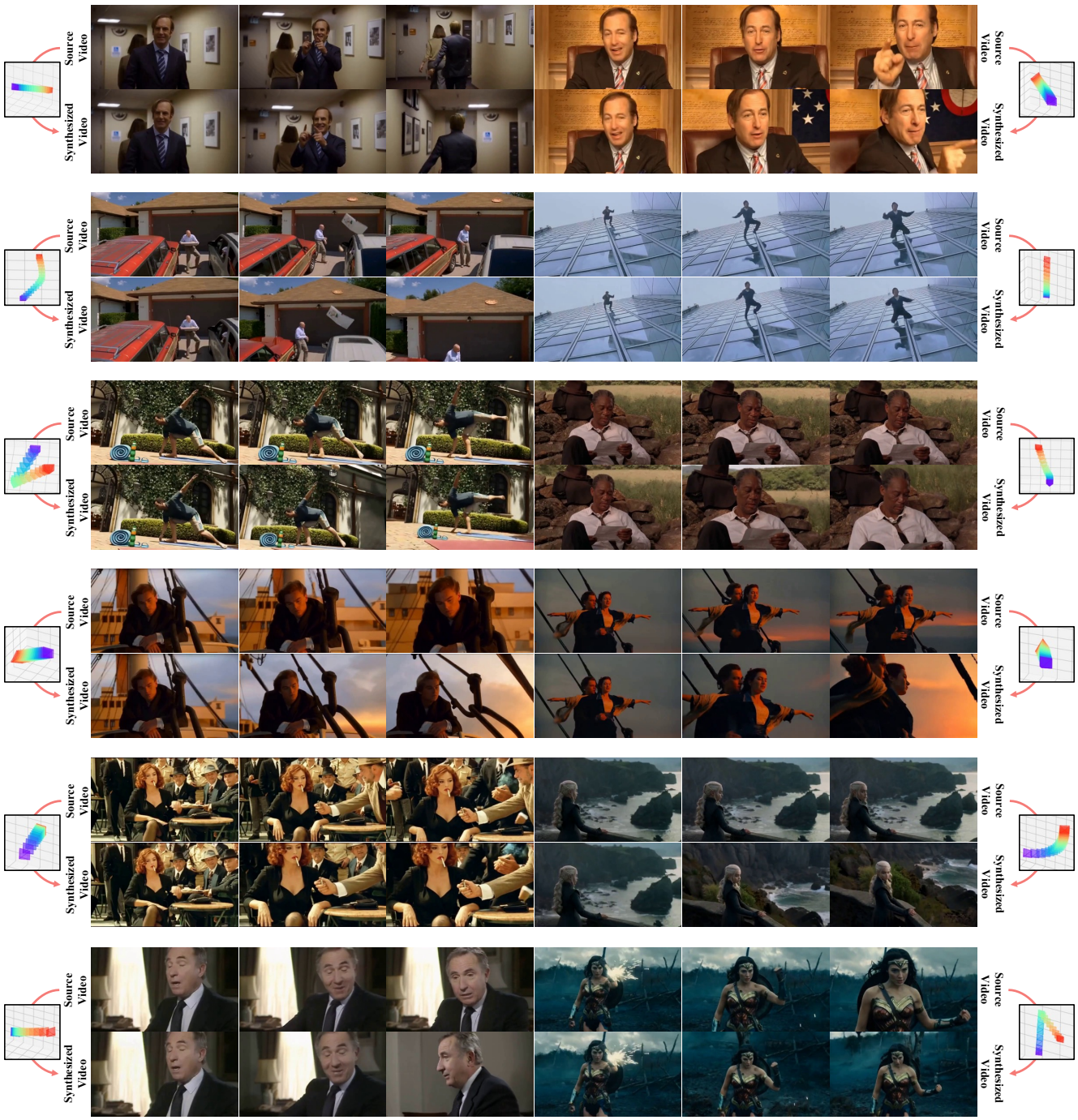

Figure 12. More synthesized results of ReCamMaster.

Figure 13. More comparison with state-of-the-art methods.