CameraCtrl II: Dynamic Scene Exploration via Camera-controlled Video Diffusion Models Appendix

Hao He[1,3*] Ceyuan Yang[3†] Shanchuan Lin[3] Yinghao Xu[5] Meng Wei[4] Liangke Gui[3]
Qi Zhao[3] Gordon Wetzstein[5] Lu Jiang[3] Hongsheng Li[1,2†]
[1]CUHK MMLab [2]CPII under InnoHK [3]ByteDance Seed [4]ByteDance [5]Stanford University

https://hehao13.github.io/Projects-CameraCtrl-II/

1. Appendix / Supplemental Material

The organization of this appendix is as follows: Sec. 2 shows some 3D reconstruction results on our camera-controlled videos. We provide some discussion regarding other related works in Sec. 3. Sec. 4 gives more details regarding the data construction process.

Sec. 5 presents more implementation details. We give some calculation details of some metrics in Sec. 6. Sec. 7 gives more ablation studies results and all the metric results for the ablations in the main paper. The details of model distillation is depicted in Sec. 8. Then, we provide more qualitative comparisons in Sec. 9. Finally, we present some discussions on limitations and present failure cases in Sec. 10.

Besides, in all visual results, the first image in each row represents the camera trajectory of a video. Each small tetrahedron on this image represents the position and orientation of a camera for one video frame. Its vertex stands for the camera location, while the base represents the imaging plane of the camera. The red arrows indicate the movement of camera **position** but do **not** depict the camera rotation. The camera rotation can be observed through the orientation of the tetrahedrons.

2. 3D Reconstruction on Generated Videos

Our method generates high-quality dynamic videos with conditional camera poses, effectively transforming video generative models into view synthesizers. The strong 3D consistency of these generated videos enables high-quality 3D reconstruction. Specifically, we use FLARE [19] to infer detailed 3D point clouds from frames extracted from our generated videos. As shown in Fig. 1, our approach produces videos that can be reconstructed into high-quality point clouds, demonstrating the superior 3D consistency achieved by our models.

3. Discussion on Other Related Work

Early works on perpetual scene generation, such as InfiniteNature [10], adopt a render, refine, repeat pipeline, using differentiable rendering [6] to autoregressively generate video frames along camera trajectories. InfiniteNature-Zero [7] eliminates the need for video datasets by solving this problem with self-supervised cycle reconstruction loss and adversarial loss. Persistent Nature [3] and DiffDreamer [2] improve 3D consistency and visual quality of generated frames, respectively. WonderJourney [17] represents a significant advantage in this field by enabling users to journey through a sequence of coherently connected scenes. Dreamscene360 [20], Wonderworld [16] and GaussianCity [15] utilize 3D Gaussian-based reconstruction and rendering methods to generate unconstrained scenes. Wonderland [8] further builds the reconstruction model in the latent space of diffusion models to make perpetual scene generation more efficient. Despite their effectiveness, these methods struggle to handle dynamic content in generated scenes and require per-scene optimization, which is time-consuming. In contrast, our method leverages pretrained video diffusion models to improve both dynamics and efficiency.

4. More Details in the Dataset Construction

After obtaining dynamic videos, we employ TMO [4] motion segmentation method to extract motion object masks. While TMO theoretically requires both optical flow and RGB images as input, we achieved satisfactory results by using RGB images for both inputs. To improve the robustness of subsequent SfM matching, we dilate the obtained masks by 5 pixels.

With the masks extracted, we utilize VGGSfM [14] (version 2.0.0) with its open-source checkpoint, code, and corresponding configuration files to estimate camera poses from video clips. To conserve GPU memory and accelerate processing, we sample frames at 4 fps from the original videos, providing both RGB images and correspond-

* Work was done during Hao He's internship at ByteDance Seed.
† Corresponding authors.

Figure 1. 3D reconstruction on generated videos by CAMERACTRL II.

ing masks to VGGSfM. After obtaining camera poses, we observed that some camera trajectories were either fragmented or exhibited shaking in certain regions. We implemented a post-processing method to filter out these problematic videos and their associated camera trajectories.

For calibration, we sample frames at 1 fps and extract metric depth using the Depth-Pro [1] model. Depth-Pro requires camera intrinsics as input to enhance accuracy, for which we use the intrinsics estimated by VGGSfM. After obtaining both metric depth and VGGSfM depth for multiple frames, we estimate a scale factor for each video scene using Equation 2 from the main paper. This scale factor is then multiplied with the translation vectors in the extrinsic matrices to obtain calibrated camera poses. Since we initially sampled frames for VGGSfM processing, we interpolate the camera poses to match the original video frame count.

We then analyzed the distribution of camera trajectory types across the dataset as described in the main paper. We found a long-tail distribution, as shown in Figure 1. To balance the dataset, we removed trajectories from over-represented categories, resulting in the more balanced distribution shown in Figure 2. Note that we processed and balanced the data in batches; the distributions shown here represent one such batch. This process yielded approximately 83K samples.

Additionally, we applied the same processing pipeline to the original RealEstate10K [21] dataset, obtaining about 37K samples. Throughout the entire pipeline, excluding initial dynamic video filtering but including camera trajectory anomaly filtering, balancing, and SfM failures, we filtered out approximately 70% of the videos. Our final dataset contains approximately 120K samples. We provide some samples of our dataset in the supplementary material.

5. More Implementation Details

Our CAMERACTRL II is built upon an internal research-focused diffusion transformer model, which shares a similar architecture with MMDiT [5]. The model consists of patchify layers, DiT blocks, and output layers, where each DiT block comprises self-attention layers, MLP, and AdaLN components. As a latent diffusion model, it employs a temporal causal VAE tokenizer similar to MAGViT2 [18], with downsampling rate 4 for temporal and 8 for spatial. This base video diffusion model is jointly trained on images and videos at 192×320 resolution. Since the temporal downsample ratio of the causal VAE is 4, we also sample the camera poses every 4 frames, resulting in the same number of camera poses to the visual features.

In implementing our camera-control model, as described in the main paper, we only introduced an additional camera patchify layer, with new parameters limited to this layer's weights and bias. During training, we kept all base video diffusion model parameters unfrozen, allowing joint optimization of all parameters.

Our training strategy involves two stages. In the first stage, we train the single-clip camera control model at 192×384 resolution, using video clips ranging from 2 to

Data Pipeline	FVD↓	Motion strength ↑	TransErr↓	RotErr ↓	Geometric consistency ↑	Appearance consistency ↑
w/o Dyn. Vid	143.28	129.40	0.2069	2.02	78.50	0.8031
w/o Scale Calib.	116.92	301.68	0.2121	2.14	82.10	0.8520
w/o Dist. Balance	**111.42**	**309.24**	0.2834	4.56	85.96	0.8500
Full Pipeline	112.46	306.99	**0.1830**	**1.74**	**86.50**	**0.8654**

Table 1. All metric results of ablation study on dataset construction process.

10 seconds in duration. The data composition maintains a 4:1 ratio between camera-labeled and unlabeled data, while Text-to-Video and Image-to-Video tasks are distributed in a 7:3 ratio. The second stage operates at an increased resolution of 384×640 to enhance output quality and incorporates video extension training. This stage maintains the same 4:1 ratio of camera-labeled to unlabeled data, with task ratios of 2:3:5 for video extension, image-to-video (single clip), and text-to-video (single clip) respectively. The joint training of video extension alongside single-clip I2V and T2V tasks represents a crucial design choice, enabling the model to learn extension capabilities while preserving single-clip camera control performance. During video extension training, the number of condition frames from the previous clip ranges from a minimum of 5 frames to a maximum of 50Both training stages utilize the AdamW optimizer. The learning rate was initially set to 1×10^{-4}, with a warm-up period from 5×10^{-5} over 500 steps. , weight decay of 0.01, and betas of 0.9 and 0.95. The learning rate was finally decayed to 1×10^{-5} using the cosine learning rate scheduler. We use 64 H100 GPUs for the first stage and 128 H100 GPUs for the second stage.

6. More Details on Metric

6.1. Details on TransErr and RotErr

The TransErr and RotErr measure the alignment between the ground truth camera trajectory and the estimated camera trajectory from the generated video clips. Similar to the dataset construction, we use the TMO [4] to extract the motion pattern of the generated vides, then use the VG-GSfM [14] to estimate the camera parameters. Since SfM is only accurate up to a relative scale, camera poses extracted from generated videos present two key challenges: 1) The camera coordinate system may have a systematic offset from the ground truth coordinate system. 2) The distances between predicted camera poses might be scaled arbitrarily. To address these issues, we align the estimated trajectory to the ground truth trajectory using the ATE [12] approach before computing error metrics. This alignment process involves:

1. Centering both trajectories by subtracting their respective means.

2. Finding the optimal scale factor between the trajectories.

3. Computing the optimal rotation between the centered trajectories using Singular value decomposition.

4. Determining the translation that aligns the trajectories.

After alignment, we calculate: **TransErr:** The average Euclidean distance between corresponding camera positions. **RotErr:** The average angular difference between corresponding camera orientations.

6.2. Details of Motion Strength

This quantitative measure calculates the average motion magnitude of foreground objects across video frames, providing insights into the model's ability to generate dynamic content. First, we extract dense optical flow fields between consecutive frames using RAFT [13]. These flow fields capture the pixel-wise displacement vectors (u, v) that represent motion between frames. To focus exclusively on object motion rather than camera movement, we utilize object segmentation masks extracted using TMO to isolate foreground regions. For each frame pair, we calculate the flow magnitude as $\sqrt{u^2+^2}$ and convert it from radians to degrees for interpretability. The motion strength value for a video is then computed as the average flow magnitude across all foreground pixels in all frames. This provides a robust measure of object motion that is independent of camera movement, allowing us to quantitatively assess the dynamic quality of generated videos.

7. More Experiment Results

7.1. Metric on the Base Video Diffusion Model

We provide the metrics of the base video diffusion model. We use the same dataset as the CAMERACTRL II for evaluating, with the resolution in 192×320. The metric for FVD, Motion strength, Geometric consistency are 310.98, 320.32. From the comparison of out model and the base video diffusion model, we conclude that the camera control training does not have negative impact on the visual quality and the dynamic. Since the base video diffusion model cannot take the camera parameters as input, we cannot calculate the TransErr, RotErr, and Geometric consistency. And the base video diffusion model are not trained for video extension, we do not have the result for Appearance consistency metric either.

Model	FVD↓	Motion strength ↑	TransErr ↓	RotErr ↓	Geometric consistency ↑	Appearance consistenct ↑
Complex Encoder	132.32	301.23	**0.1826**	1.88	84.00	**0.8760**
Multilayer Inj.	128.53	247.23	0.1865	1.78	85.00	0.8210
w/o Joint Training	122.10	279.82	0.2098	1.97	81.92	0.8400
CAMERACTRL II	**112.46**	**306.99**	0.1830	**1.74**	**86.50**	0.8654

Table 2. All metric results of ablation study on the impact of our model architecture and training strategy for single-clip camera-controlled video generation.

Model	FVD↓	Motion strength ↑	TransErr ↓	RotErr ↓	Geometric consistency ↑	Appearance consistency ↑
Different Ref.	118.32	303.66	0.1963	1.94	**87.37**	0.8032
Noised Condition	136.78	306.21	0.1847	1.85	83.87	0.7843
CAMERACTRL II	**112.46**	306.99	**0.1830**	**1.74**	86.50	**0.8654**

Table 3. All metric results of ablation study on key design choices in extending the single-clip model to enable scene exploration.

Model	FVD ↓	Motion strength ↑	TransErr ↓	RotErr ↓	Geometric consistency ↑	Appearance consistennncy ↑
Scale on Time Embedding	132.32	303.64	0.1993	1.82	85.00	0.8070
Camera CFG'	114.20	298.42	0.1915	1.79	84.20	**0.8690**
Learnable Null Plücker	**110.32**	287.82	0.2098	1.91	**87.90**	0.8320
Noised Condition*	140.98	303.76	0.1901	1.88	82.90	0.7982
CAMERACTRL II	112.46	**306.99**	**0.1830**	**1.74**	86.50	0.8654

Table 4. Extra ablation studies.

7.2. Comprehensive Metric Results

Due to the constraint of the space, in the main paper, we only include the highly related metric results for the ablation experiments. Here, we provide the results of all metrics in Tab. 1, Tab. 2, and Tab. 3.

7.3. More Ablation Study

We provide more ablation studies in Tab. 4. First, we explore a different approach for incorporating scale information. Instead of directly calibrating camera poses, we experiment with injecting scale factors as timestep-like embedding while using uncalibrated camera poses (Tab. 4 Scale on Time Embedding). Although this theoretically allows the model to learn the scale normalization to metric space, it shows degraded in the camera control accuracy, This suggests that providing the model with the normalized scene scale directly helps the model better learn geometric relationship, rather than requiring it to implicitly learn the unified scene scale.

We then explore different strategies for camera classifier-free guidance during inference. Besides the approach described in Equ. 3 of the main paper, we evaluate an alternative formulation

$$\begin{aligned}\hat{\epsilon}_\theta(z_t, c, s, t) = \epsilon_\theta(z_t, \phi_{text}, \phi_{cam}) \\ + w_{text}(\epsilon_\theta(z_t, c, \phi_{cam}) - \epsilon_\theta(z_t, \phi_{text}, \phi_{cam})) \\ + w_{cam_1}(\epsilon_\theta(z_t, \phi_{text}, s) - \epsilon_\theta(z_t, \phi_{text}, \phi_{cam})) \\ + w_{cam_2}(\epsilon_\theta(z_t, c, s) - \epsilon_\theta(z_t, c, \phi_{cam})) \quad (1)\end{aligned}$$

Data Pipeline	FVD↓	TransErr↓	RotErr ↓	Sample time (s) ↓
Before distillation	**73.11**	**0.1892**	**1.66**	13.83
Progressive distill	86.32	0.2001	1.90	2.61
APT	198.21	0.2500	2.56	**0.59**

Table 5. Model comparison before and after the distillation. The inference time is tested when generating 4 second 12fps video with 4 H800 GPU.

As shown in Tab. 4 Camera CFG', this alternative approach does not improve camera pose control capabilities. Besides, it will increase the inference time. Therefore, we opt for the original formulation presented in the main paper.

After that, in the multi-clip video extension, our initial choice on adding noise on the previous clips during training leads to misalignment between training and inference thus results in degraded performance (Tab. 3). Here, we investigate whether adding small noise on the previous clips during the inference will compensate for this gap, results are shown in Tab. 4 Noised Condition*. It turns out that this strategy does not improve the performance.

Finally, we examine an alternative approach for handling unlabeled data: using a learnable null embedding. The results in Tab. 4 Learnable Null Plücker demonstrate that the learnable embedding does not improve camera control metrics. Our chosen approach of using zero tensors is more intuitive and potentially reduces model learning complexity compared to the learnable alternative.

8. Details of Model Distillation

Our model employs two distinct classifier-free guidance mechanisms for text and camera control. Without any form of distillation, our inference process requires 96 (32×3) neural function evaluations (NFE). To accelerate inference and improve user experience, we implemented a two-phase distillation approach. First, we employed progressive distillation [11] to reduce the required NFEs from 96 to 16 while maintaining visual quality. As shown in the first two row of Tab. 5, the distilled model does not exist significant degradation in terms of visual quality and camera control accuracy. When generating a 4 second videos in 12fps with 4 H800 GPUs, the sample time is decreased significantly, from 13.83 second to 2.61 second. This sample time contains the DiT model inference time and the VAE decode time.

To further accelerate, we applied the recent distillation method APT [9] to distill the model to single step. As shown in the last row of Tab. 5, this single step model offers the fastest generation speed, enabling near real-time video creation with only a modest reduction in visual quality and camera control.

9. More Qualitative Comparisons

We provide more qualitative comparisons in Fig. 6 with CAMERACTRL II and the state-of-the-art camera control model AC3D. Compared to AC3D, our model can strictly follow the input camera trajectory input, and generate the dynamic videos. For example, for the first camera trajectory, AC3D fails to generate the left movement at the end of the camera trajectory. For the second example, our model generates the camera movement and turn successfully, while AC3D continues to generate the leftward movement. For the last example, the camera trajectory shows camera turn left in a large degree (near 90 degree). Our model can generate a video follows such a trajectory, while AC3D fails to generate a video in such large camera turn.

10. Limitations and Failure Cases

Limitation and Future Work. Our current approach has several limitations for future investigation. First, CAMERACTRL II occasionally struggles to resolve conflicts between camera movement and scene geometry, sometimes resulting in physically implausible camera paths that intersect with scene structures. Additionally, while our method achieves accurate camera control, the overall geometric consistency of generated scenes could be further improved, especially when dealing with complex camera trajectories.

Failure Cases. While our method demonstrates strong performance in most scenarios, we also identify limitations in handling conflicts between camera trajectories and physical scene constraints. Fig. 5 illustrates such a failure case that aligns with the discussion in our main paper. In this example, we provide a forward camera trajectory when left and right camera view change. There a fence blocks the intended path. An ideal physically-aware model would recognize this constraint and stop the camera movement at the fence. However, our model generates a physically implausible result where the fence structure deteriorates as the camera passes through it. We believe this limitation stems from the video generation model's incomplete understanding of physical constraints in the world. Current video diffusion models primarily learn appearance and motion patterns from data without explicit physical reasoning capabilities. This physical awareness challenge represents an important direction for future research in camera-controlled video generation, potentially bridging the gap between visual generation and physical simulation.

References

[1] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. 2

[2] Shengqu Cai, Eric Ryan Chan, Songyou Peng, Mohamad Shahbazi, Anton Obukhov, Luc Van Gool, and Gordon Wetzstein. Diffdreamer: Towards consistent unsupervised single-view scene extrapolation with conditional diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2139–2150, 2023. 1

[3] Lucy Chai, Richard Tucker, Zhengqi Li, Phillip Isola, and Noah Snavely. Persistent nature: A generative model of unbounded 3d worlds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20863–20874, 2023. 1

[4] Suhwan Cho, Minhyeok Lee, Seunghoon Lee, Chaewon Park, Donghyeong Kim, and Sangyoun Lee. Treating motion as option to reduce motion dependency in unsupervised video object segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5140–5149, 2023. 1, 3

[5] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 2

[6] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8377–8386, 2018. 1

[7] Zhengqi Li, Qianqian Wang, Noah Snavely, and Angjoo Kanazawa. Infinitenature-zero: Learning perpetual view generation of natural scenes from single images. In *European Conference on Computer Vision*, pages 515–534. Springer, 2022. 1

[8] Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N Plataniotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. *arXiv preprint arXiv:2412.12091*, 2024. 1

[9] Shanchuan Lin, Xin Xia, Yuxi Ren, Ceyuan Yang, Xuefeng Xiao, and Lu Jiang. Diffusion adversarial post-training for one-step video generation. *arXiv preprint arXiv:2501.08316*, 2025. 5

[10] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[11] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 5

[12] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 3

[13] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 3

[14] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024. 1, 3

[15] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. Gaussiancity: Generative gaussian splatting for unbounded 3d city generation. *arXiv preprint arXiv:2406.06526*, 2024. 1

[16] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. *arXiv preprint arXiv:2406.09394*, 2024. 1

[17] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6658–6667, 2024. 1

[18] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 2

[19] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. *arXiv preprint arXiv:2502.12138*, 2025. 1

[20] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In *European Conference on Computer Vision*, pages 324–342. Springer, 2024. 1

[21] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 2

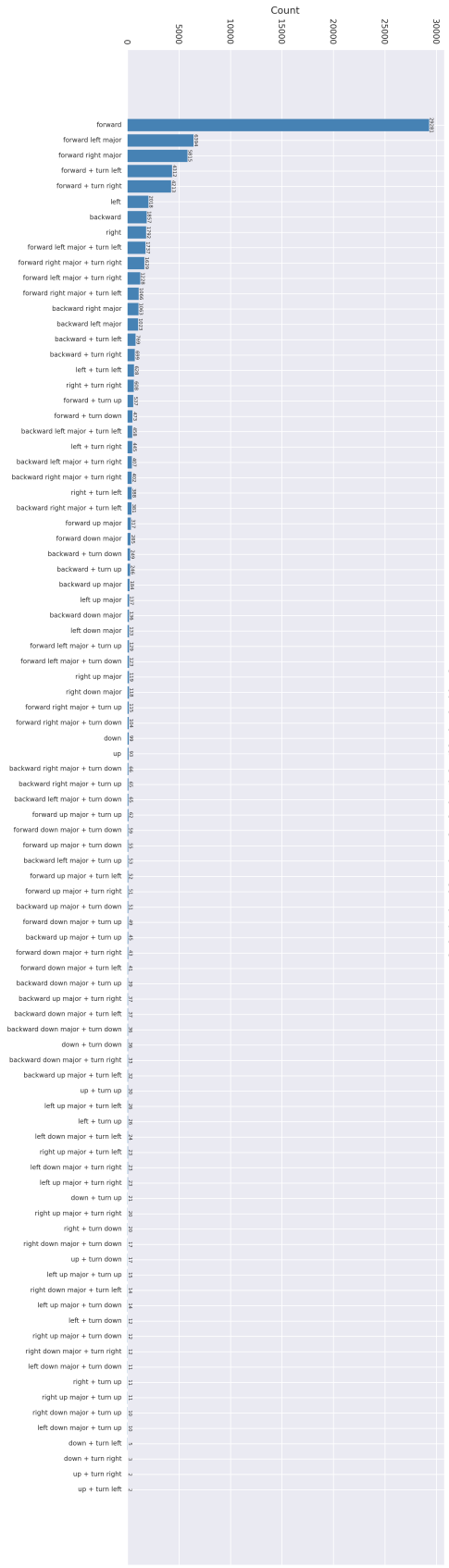

Figure 2. Camera trajectory type distribution before the dataset balancing

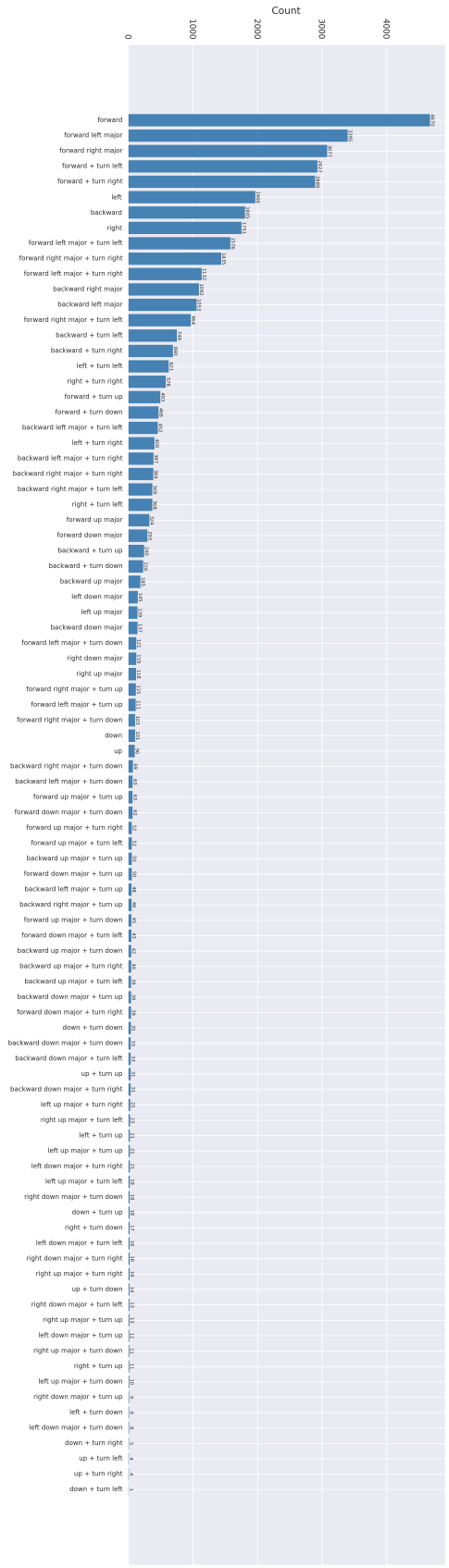

Figure 3. Camera trajectory type distribution after the dataset balancing

Figure 4. Visualization results between our models before and after the distillation. We generate these videos in the I2V setting, with the first image as the condition images.

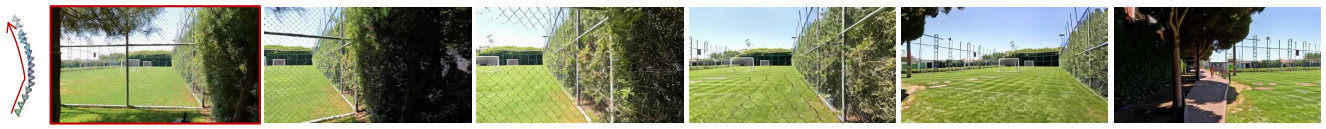

Figure 5. Visualization results of a failure case. We generate this video in the I2V setting, with the first image as the condition image.

Figure 6. **Qualitative comparisons**. Rows 1, 3, 5 give the results of AC3D, results of CAMERACTRL II are shown in rows 2, 4, 6.