# CameraAnything: Refilming Videos with Arbitrary Camera Control

Yixuan Li[1,2✉], Yanhong Zeng[2,3], Ka Leong Cheng[2], Jiayi Zhu[2], Hanlin Wang[2,5], Wen Wang[2,4], Yihao Meng[2,5], Hao Ouyang[2], Qiuyu Wang[2], Yue Yu[2,5], ZiDong Wang[1], Yiyuan Zhang[1], Yujun Shen[2], Dahua Lin[1]

[1] The Chinese University of Hong Kong, Hong Kong, China
liyixxxuan@gmail.com
[2] Ant Group, Hangzhou, China
zengyh1900@gmail.com
[3] Tsinghua University, Beijing, China
[4] Zhejiang University, Hangzhou, China
[5] The Hong Kong University of Science and Technology, Hong Kong, China

https://yixuanli98.github.io/cameraanything/

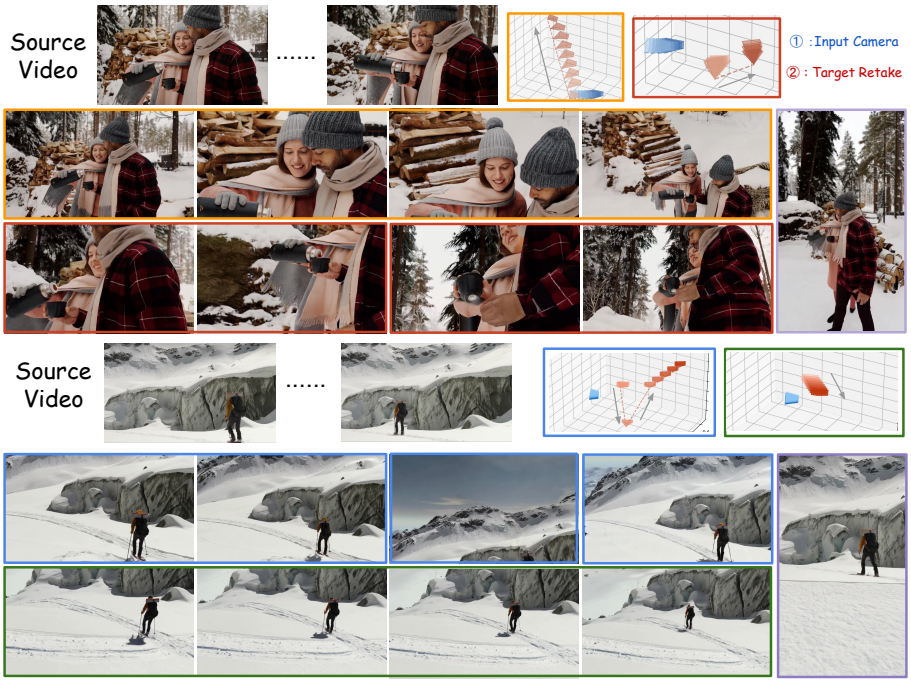


**Fig. 1:** We propose **CameraAnything**, the first unified video editing framework that enables comprehensive camera control, including novel trajectory manipulation, multi-shot transitions, resolution adaptation, and focal length adjustment in a single framework, demonstrating strong potential for cinema-level video editing.

Y. Li, Y. Zeng—Equal contributions.

**Abstract.** We introduce **CameraAnything**, the first unified framework for camera controlled video editing that enables joint control of both intrinsic and extrinsic camera parameters. Existing approaches either rely on expensive 3D reconstruction to achieve full camera functionality or restrict editing to extrinsic parameter manipulation. Moreover, the coupled influence of intrinsic and extrinsic parameters on video appearance makes disentangled modeling particularly challenging. To address this, we adopt per-pixel Plücker ray injection alongside resolution-aware 3D RoPE in self-attention, building both camera conditioning and spatial positional encoding on the target latent to jointly control camera position, focal length, and native resolution editing without cropping or outpainting. To overcome the scarcity of paired training data, we further develop a scalable synthetic pipeline that constructs diverse dynamic scenes through structured multi-camera recording and generates synchronized videos with varied camera configurations. With a tailored orthogonal training strategy, CameraAnything enables expressive video reshooting with arbitrary viewpoint control, focal length adjustment, resolution adaptation, and multi-shot transitions within a single generation process, offering strong practical value for cinematic video editing and cross-platform content adaptation in video production.

**Keywords:** Camera-controlled video editing · Diffusion models

## 1 Introduction

Controlling the camera is fundamental to visual storytelling. Recent advances in generative video models enable controllable camera movements, allowing users to reshoot existing videos with novel camera motions [2, 7, 23]. Such capabilities promise to democratize cinematic video production by enabling complex camera effects without requiring professional equipment or multi-camera setups.

Expressive camera language arises from the joint manipulation of **extrinsic** parameters (camera position and orientation) and **intrinsic** parameters (e.g., focal length and resolution), which determine scene projection, framing, and viewer attention. Despite promising progress, state-of-the-art camera-controlled video editing methods support only extrinsic pose control and lack intrinsic parameter manipulation such as focal length, limiting their ability to reproduce expressive camera language [2, 7, 23]. Moreover, many methods assume fixed resolutions or aspect ratios, restricting applicability across diverse display platforms.

In this paper, we present **CameraAnything**, the first unified framework for camera-controlled video editing that supports expressive camera language. As illustrated in Fig. 1, it enables four forms of camera control: viewpoint manipulation, focal length adjustment, resolution adaptation, and multi-shot transitions within a single generation process. Given an input video and target camera parameters, CameraAnything conditions a video diffusion transformer on camera-aware geometric representations to generate videos that follow the specified configuration. A key challenge is modeling camera parameters that jointly affect scene appearance, as intrinsic and extrinsic parameters are often tightly coupled

in cinematic operations. For example, the Hitchcock dolly zoom simultaneously changes focal length and camera position to maintain the subject's apparent size while dramatically altering background perspective [1].

To address this problem, we adopt per-pixel Plücker ray injection alongside resolution-aware 3D RoPE in self-attention, building both camera conditioning and spatial positional encoding on the target latent. We represent camera geometry using token-wise Plücker rays [26], which encode the 3D viewing direction and moment of each pixel ray, uniquely representing the camera ray associated with each image token. This formulation naturally captures both intrinsic and extrinsic camera parameters. Following Rotary Camera Encoding (RoCE) [23], we inject these embeddings into Rotary Position Encoding (RoPE) within the transformer attention layers [28], augmenting the positional encoding used for query–key interactions. For native resolution editing, we directly recompute the spatial coordinates of 3D RoPE from the target-resolution latent grid, so token positional encoding adapts to the desired output size. By grounding visual tokens in camera geometry while preserving RoPE's positional generalization, the model learns geometry-consistent representations that remain stable under diverse camera configurations. In practice, we evaluate several representation and injection variants through ablation and adopt the configuration above as the most effective for unified camera control. Training such a model requires paired videos with controlled camera variations, which are difficult to collect in the real world. We build a scalable synthetic data pipeline based on structured multi-camera recording, rendering synchronized clips per scene with diverse extrinsic trajectories, focal changes, and aspect-ratio variations. We further employ an orthogonal training strategy that independently samples extrinsic, focal, and resolution edits, enabling both single-dimension control and compound camera instructions.

Extensive experiments demonstrate that CameraAnything achieves state-of-the-art performance across multiple camera-controlled video editing tasks, achieving expressive camera language and facilitating practical video adaptation across different display formats. Our contributions are summarized as follows:

- We introduce **CameraAnything**, a unified framework for camera-controlled video editing that jointly manipulates viewpoint, focal length, resolution, and multi-shot transitions within a single generation process. To realize these functions, we inject per-pixel Plücker camera embeddings alongside resolution-aware 3D RoPE on the target latent, as selected through ablation over alternative representation and injection designs.
- We build a scalable synthetic data pipeline using structured multi-camera recording to generate synchronized training videos with diverse and controllable camera configurations, and design an orthogonal training strategy that independently samples extrinsic, focal, and resolution edits to support both single-dimension control and compound camera instructions.
- Extensive experiments demonstrate state-of-the-art performance on camera-controlled video editing tasks and highlight the framework's potential for cinematic video production and cross-platform content adaptation.

## 2 Related works

*Camera-controlled video generation.* Camera-controlled video generation has attracted growing research interest for its potential to enable film-level narrative storytelling and democratize individual film production. A central challenge in this area is how to effectively represent and inject camera parameters into video generation models. Early approaches adopt explicit camera parameterizations such as Plücker embeddings or extrinsic matrices. CameraCtrl [9] introduces a camera encoder that maps Plücker embeddings [26] into multi-scale features, which are integrated into all temporal attention layers of a U-Net–based video model. CameraCtrl-II [10] extends this by curating a dataset with extensive dynamics and camera annotations, injecting camera representations via element-wise addition before the first DiT layer to improve viewpoint range and motion diversity. SynCamMaster [3] directly uses camera extrinsic matrices and trains on Unreal Engine–rendered multi-camera videos with a hybrid training scheme to support multi-camera generation. ShotDirector [33] combines both representations through a dual-branch design with an extrinsic branch and a Plücker branch, enabling multi-shot generation with flexible shot transitions. More recent works explore tighter integration of 3D geometry with visual tokens in Transformer architectures, including token-level raymap encodings [8, 26], attention-level relative pose encodings [18, 22], and relative projective positional encoding (PRoPE) with full frustum representations [16, 19, 37].

*Camera-controlled video editing.* While video editing has been extensively studied for tasks such as style transfer, local editing, and inpainting [17, 39], camera-controlled video editing, which aims to re-render an existing video from novel viewpoints, remains comparatively underexplored. One line of work reconstructs explicit 3D representations (e.g., point clouds, meshes) from reference videos and then re-renders them from novel camera trajectories, often followed by quality enhancement to handle artifacts and enable additional controllability such as focal length and illumination adjustments [5, 11, 12, 20, 25, 27, 35]. However, these methods are constrained by the fidelity of 3D reconstruction, particularly for complex or dynamic scenes. Another line of work directly trains generative models to re-shoot videos with new camera trajectories. ReCamMaster [2], a pioneering effort in this direction, trains on large-scale multi-camera synthetic synchronized videos and uses camera extrinsic matrices to condition generation. ReDirector [23] proposes RoCE, which encodes camera information through relative positional embeddings in the spatial component of 3D-RoPE within self-attention layers. PlenopticDreamer [7] synchronizes generative hallucinations across multi-view video re-rendering by maintaining spatio-temporal memory through autoregressive generation with camera-guided video retrieval. Beyond static viewpoint changes, recent works further explore joint camera control with additional conditions, such as world time [14, 32], professional photographic parameters including bokeh, exposure, and color tone [29, 34, 36], or replicating camera movements from reference videos without requiring explicit camera parameters [21]. However, existing methods are generally limited to reshooting

videos with smooth, continuous extrinsic camera pose transitions along a single trajectory, without supporting intrinsic camera parameter adjustments. In contrast, our work presents the first unified framework that enables both multi-shot camera pose changes within a single trajectory and intrinsic parameter adjustments including focal length and resolution, offering significant practical value for adapting content across different display devices, such as portrait mode for mobile screens and horizontal formats for television broadcasts.

## 3 Dataset

To support our proposed framework—the first to enable re-shooting single-shot videos with novel trajectories, multi-shot editing, and intrinsic control—a specialized dataset is required. While ReCamMaster [2] provides a foundation for modifying trajectories, it is limited by static camera intrinsics and continuous trajectories, failing to support re-focal effects or the transition from a single long take to a cinematic multi-shot edit. Consequently, we develop a high-fidelity synthetic dataset using Unreal Engine 5 [6], enabling joint control over camera extrinsics $(R, T)$ and intrinsics, including focal length $f$ and resolution $(H, W)$.


**Fig. 2:** Dataset Example. The red box represents single-shot settings with consistent camera extrinsics, including both fixed and varying focal lengths. The orange and blue boxes each represent a different shot in multi-shot sequences.

We construct diverse 3D environments populated with animated human characters to simulate real-world dynamic scenes. To enable the model to capture the spatial correlations between alternative viewpoints and the temporal continuity across cinematic cuts, we employ a structured recording protocol. For

each scene, we render 20 synchronized video clips, strictly aligned in the time dimension but varying in camera and editing configurations. The 20 clips per scene are organized into three distinct groups to facilitate multi-task learning:

- Group A (Base Views, Clips 1-5): These clips are rendered as continuous, single-shot trajectories with a standard fixed focal length. They serve as the raw input "anchors" providing the base visual and motion information.
- Group B (Multi-shot Editing Variants, Clips 6-10): These clips replace smooth trajectories with **1–3** instantaneous camera cuts per video, where the camera undergoes abrupt position and orientation changes to simulate cinematic shot transitions. By pairing Group-A single-shot sources with these multi-shot targets, the model learns to transform a continuous long take into an edited sequence with distinct shots.
- Group C (Re-focal Variants, Clips 11–20): Unlike prior frameworks that keep camera intrinsics fixed, we construct this group to enable re-focal capability. These clips reuse the exact extrinsic trajectories from Groups A and B but are rendered with different focal lengths. By pairing clips with identical camera motion yet varying fields of view (FoV), the model learns to disentangle geometric camera motion from optical zooming effects, enabling more flexible focal length control during generation.

Beyond focal length, we introduce a strategy to achieve native resolution and aspect ratio editing, which we treat as critical components of camera intrinsics. We render all raw footage in a high-resolution square format $(1:1)$ to serve as a versatile data source. During training, we perform dynamic, parameterized center cropping on these square frames to generate training samples. This approach allows the model to learn how to re-compose the scene for various output formats (e.g., 16:9, 9:16) and resolutions while maintaining visual fidelity—a capability entirely absent in previous fixed-intrinsic frameworks. We provide some cases in Fig. 2. Please refer to the supplementary material for more dataset details.

## 4 CameraAnything

### 4.1 Preliminary

*Video Diffusion Transformer.* As illustrated in Fig. 3, we build on Wan2.1-T2V-1.3B [31], a latent video diffusion model that encodes frames into a compact latent space using a 3D Variational Autoencoder (VAE) and denoises them with a Diffusion Transformer (DiT). The 3D-VAE compresses a video $V \in \mathbb{R}^{F\times H\times W\times 3}$ into a latent $z \in \mathbb{R}^{F'\times H'\times W'\times C}$, with temporal and spatial strides of 4 and 8 respectively. The DiT processes $N = F' \times H' \times W'$ spatiotemporal tokens obtained by spatially patchifying the latent. Each DiT block applies timestep-conditioned adaptive layer normalization (AdaLN) before self-attention and feed-forward layers, while 3D Rotary Positional Embeddings (RoPE) encode the spatiotemporal coordinates $(f, h, w)$ of each token in the attention computation.

For video-to-video generation, we follow ReCamMaster [2] and concatenate the source video tokens $x_s$ and target video tokens $x_t$ along the frame dimension,

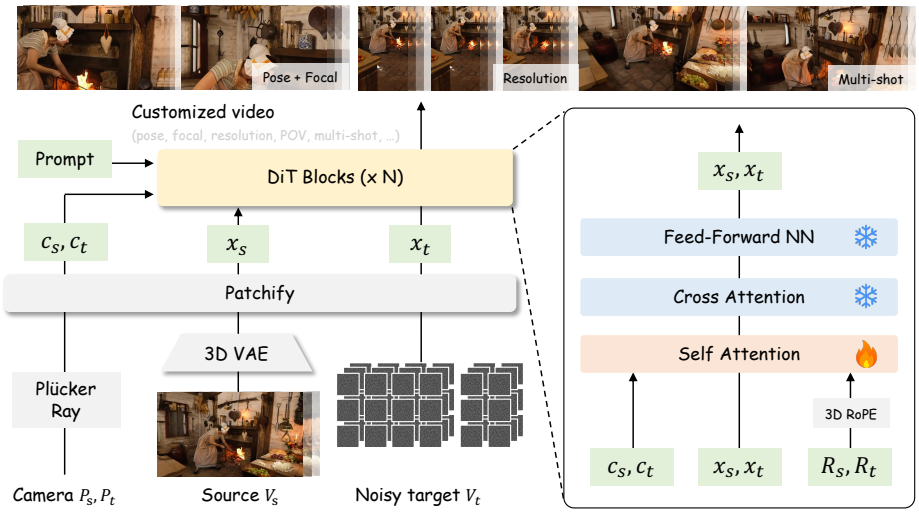


**Fig. 3: Overview of CameraAnything.** CameraAnything is built upon Wan2.1-T2V-1.3B [31], which includes a 3D VAE for video latent encoding and a diffusion Transformer composed of multiple DiT blocks. Given a source video together with its camera parameters and a target camera specification, CameraAnything represents cameras using Plücker rays to provide accurate per-token geometric modeling, which is injected into the self-attention modules. With this design, CameraAnything can generate videos under diverse camera edits, including novel camera poses, focal length adjustments, resolution adaptation, and multi-cut shot transitions, while also supporting flexible combinations of these controls within a single generation process.

forming a joint sequence of length $2N$ as input to the DiT. This *frame-dimension conditioning* enables direct spatiotemporal attention between source and target tokens through every self-attention layer, yielding stronger content consistency than channel-dimension or view-dimension alternatives [2].

*Camera Parameterization.* A camera at frame $i$ is described by its camera-to-world (c2w) extrinsic matrix $\mathbf{P}_i = [\mathbf{R}_i \mid \mathbf{t}_i] \in \mathbb{R}^{3\times4}$ and intrinsic matrix $\mathbf{K}_i \in \mathbb{R}^{3\times3}$. Following standard pinhole optics, we derive $\mathbf{K}_i$ from the horizontal field-of-view:

$$f_x = \frac{W}{2\tan(\text{hfov}/2)}, \quad \mathbf{K}_i = \begin{pmatrix} f_x & 0 & W/2 \\ 0 & f_x & H/2 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

All camera poses are expressed relative to the first frame of the source video,

$$\hat{\mathbf{P}}_i = \mathbf{P}_{\text{src},0}^{-1} \cdot \mathbf{P}_i, \tag{2}$$

so that $\hat{\mathbf{P}}_0 = \mathbf{I}$ and subsequent poses encode viewpoint changes in a coordinate-system-agnostic manner.

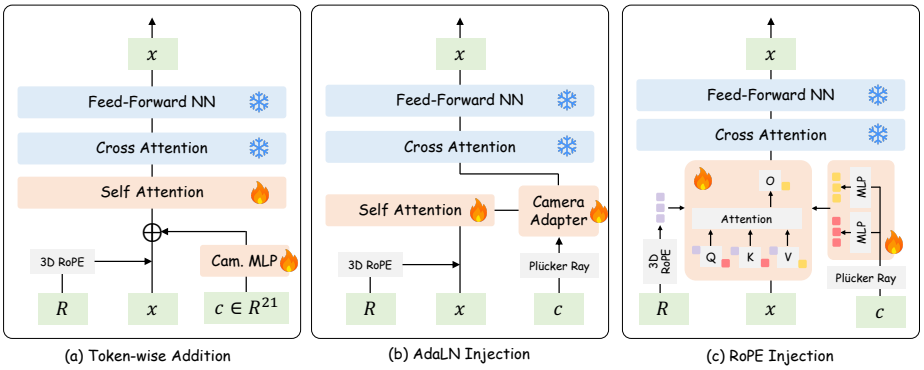


**Fig. 4:** Illustration of three camera representation and injection mechanisms. **(a) Token-wise Addition**: $\mathbf{P}_i \in \mathbb{R}^{3\times4}$ and $\mathbf{K}_i \in \mathbb{R}^{3\times3}$ are flattened and concatenated into $\mathbf{c}_i \in \mathbb{R}^{21}$, then linearly encoded and added to all tokens. **(b) AdaLN Injection**: per-pixel camera embeddings based on the Plücker ray representation are encoded by a learnable linear layer to generate scale and shift parameters that modulate the Layer Normalization layers. **(c) RoPE Injection**: embedded Plücker rays are mapped to phase shifts that act as positional encodings, applied in parallel with the original 3D RoPE during self-attention computation. Red flames indicate trainable components, while snowflakes denote frozen weights.

### 4.2 Camera Representation and Injection

A central design question is how to encode the target camera trajectory $\{\hat{\mathbf{P}}_i, \mathbf{K}_i\}_{i=0}^{F-1}$ and inject it into the DiT to steer novel-view synthesis. We systematically study two camera representations paired with three injection mechanisms (Fig. 4), and identify the combination that best supports our unified control framework.

*Linear Camera Representation.* Following ReCamMaster [2], each frame's camera is encoded by flattening the extrinsic matrix $\mathbf{P}_i = [\mathbf{R}_i \mid \mathbf{t}_i] \in \mathbb{R}^{3\times4}$ and intrinsic matrix $\mathbf{K}_i \in \mathbb{R}^{3\times3}$, then concatenating them into a 21-dimensional vector $\mathbf{c}_i = [\mathrm{vec}(\mathbf{P}_i); \mathrm{vec}(\mathbf{K}_i)]$. A learnable linear encoder $\mathcal{E}_{\mathrm{cam}} : \mathbb{R}^{21} \to \mathbb{R}^d$ projects $\mathbf{c}_i$ to dimension $d$ and broadcasts the embedding to all spatial tokens. This frame-level representation is lightweight but lacks per-pixel geometric detail.

*Plücker Ray Representation.* To obtain per-pixel geometric detail, we adopt Plücker ray embeddings [26], following CameraCtrl [9] and BulletTime [32]. For each pixel $(u, v)$ in frame $i$, the world-space ray direction and camera origin are:

$$\mathbf{d}_{i,uv} = \mathbf{R}_i \cdot \hat{\mathbf{d}}_{i,uv}, \quad \hat{\mathbf{d}}_{i,uv} = \left[\tfrac{u-c_x}{f_x},\ \tfrac{v-c_y}{f_x},\ 1\right]^\top / \|\cdot\|, \quad \mathbf{o}_i = \mathbf{t}_i. \tag{3}$$

The per-pixel embedding $\boldsymbol{\pi}_{i,uv} = [\mathbf{o}_i; \mathbf{d}_{i,uv}] \in \mathbb{R}^6$ encodes both viewpoint and focal length at each spatial location, yielding a feature map $\boldsymbol{\Pi} \in \mathbb{R}^{F\times H\times W\times 6}$. After patchification, a learned projection maps these per-token embeddings to

the hidden dimension $d$ and makes them available to every DiT block. This representation also makes resolution editing native to the camera model. Given a target resolution $H_t \times W_t$ and field-of-view, the intrinsic matrix $\mathbf{K}_t$ determines the ray direction for every pixel. When the resolution changes, the ray directions, principal point, and number of rays change accordingly, so the Plücker ray field naturally adapts to the new imaging geometry instead of relying on cropping, masks, or outpainting.

*Token-wise Addition.* The simplest injection strategy directly adds the projected camera embedding to the hidden state prior to self-attention:

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathcal{E}_{\text{cam}}(\mathbf{c}), \tag{4}$$

where $\mathbf{c}$ denotes the camera embedding for the current token [2, 9]. Although straightforward and zero-initializable, additive injection directly modifies the hidden-state distribution of the pretrained backbone, which can introduce training instability when optimizing multiple control dimensions simultaneously.

*AdaLN Injection.* Adaptive Layer Normalization (AdaLN) instead uses the camera embedding to predict per-token scale and shift parameters [10, 32]:

$$\tilde{\mathbf{x}} = (1 + \boldsymbol{\gamma}) \odot \mathbf{x} + \boldsymbol{\beta}, \quad [\boldsymbol{\gamma}, \boldsymbol{\beta}] = \text{MLP}\big(\mathcal{E}_{\text{cam}}(\mathbf{c})\big), \tag{5}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are zero-initialized so the injection begins as identity. AdaLN modulates feature distributions without directly overwriting token content, providing smoother incorporation of the camera signal while preserving the pretrained model's generative prior.

*RoPE Injection.* An orthogonal design integrates camera geometry directly into the attention mechanism. ReDirector [23] proposes Rotary Camera Encoding (RoCE), which augments the spatial components of the standard 3D RoPE with learned, Plücker-conditioned phase shifts $\boldsymbol{\phi}$:

$$\bar{\mathbf{q}}' = \bar{\mathbf{q}} \circ \mathbf{R}_{\text{rope}} \circ e^{i\boldsymbol{\phi}}, \quad \bar{\mathbf{k}}' = \bar{\mathbf{k}} \circ \mathbf{R}_{\text{rope}} \circ e^{i\boldsymbol{\phi}}, \tag{6}$$

where $\boldsymbol{\phi}$ is predicted by a lightweight MLP and zero-initialized for stable fine-tuning. Encoding camera information as a phase shift in the complex rotation domain naturally modulates the attention similarity: tokens sharing consistent viewpoints receive boosted attention while attention to geometrically distant tokens is suppressed. The original 3D RoPE is also resolution-aware: its spatial coordinates are computed from the patchified latent grid $(h', w')$, where $h' = H_t/(s_{\text{vae}} \cdot s_{\text{patch}})$ and $w' = W_t/(s_{\text{vae}} \cdot s_{\text{patch}})$. Changing the output resolution directly changes this latent grid, allowing RoPE frequencies to adapt to the new spatial layout without architectural changes.

*Design Choice.* Our ablation (Sec. 5.3) shows that Plücker rays with RoPE injection achieve the best reconstruction fidelity. Per-pixel Plücker embeddings capture spatially varying camera geometry beyond ReCamMaster's frame-level $\mathbb{R}^{21}$ encoding, and RoPE injection incorporates multi-view relationships without perturbing pretrained features. At inference time, generating a video at a new target resolution only requires constructing $\mathbf{K}_t$ from the desired $(H_t, W_t, \text{hfov}_t)$, computing per-pixel Plücker rays from $\mathbf{K}_t$ and the target extrinsic poses, and running the DiT on the corresponding latent grid. No retraining, padding, masks, or post-processing are required.

### 4.3 Training Strategy

We decompose our video-to-video generation task into three orthogonal control dimensions: (1) camera extrinsics (motion/cut), (2) focal length, and (3) aspect ratio. During training, each sample is constructed by independently sampling along these three axes, enabling the model to learn each control dimension as well as their arbitrary combinations.

Specifically, for each training iteration, we first sample the extrinsics type from same-camera, different single-shot, single-to-multi-shot with probabilities 0.2, 0.4, 0.4. The same-camera mode keeps identical camera trajectory between the condition and target videos, while different single-shot draws two distinct single-shot cameras from the same scene, and single-to-multi-shot pairs a single-shot condition with a multi-shot target. We then independently decide whether to apply a focal length change (probability 0.5) and an aspect ratio change (probability 0.2). When focal change is activated, the target video is drawn from a scene rendered with a different focal length while maintaining the same camera extrinsics, allowing the model to disentangle intrinsic and extrinsic camera parameters. For aspect-ratio and resolution editing, the target camera rays and latent grid are constructed at the target output resolution, so the model learns native resolution adaptation rather than fixed-resolution cropping or outpainting. To ensure every training sample carries a meaningful learning signal, we enforce that at least one dimension must differ between the condition and target; if the sampled configuration yields no change (same camera, no focal change, no aspect ratio change), we promote it to a focal-change sample.

This orthogonal decomposition yields up to $3\times2\times2 = 12$ task combinations, covering both single-dimension control (changing only extrinsics, focal length, or aspect ratio) and compound control instructions (e.g., simultaneously changing the camera trajectory, focal length, and aspect ratio).

## 5 Experiments

### 5.1 Experiment Settings

*Implementation details.* Our model is built upon Wan2.1-T2V-1.3B [31], with only the self-attention layers, the camera encoder, and the camera adapter modules set as trainable; all remaining backbone parameters are kept frozen. For

each training sample, the source video $V_s$ is always drawn from the single-shot, fixed-focal subset of our dataset (Sec. 3), while the target video $V_t$ is selected according to the sampled task dimensions (extrinsic change, focal change, or resolution change) as described in Sec. 4. The resolution of each sample is drawn from a predefined bucket set {480×832, 832×480, 384×672, 672×384, 512×672, 672×512, 384×512, 512×384, 640×640, 320×320}; when the resolution-change dimension is active, source and target resolutions are sampled independently to guarantee an aspect-ratio difference. The newly introduced camera modules are zero-initialized prior to training. We train for 20k steps with a learning rate of $5\times10^{-5}$ and a batch size of 32.

*Baselines.* We evaluate against two groups of methods aligned with the two task categories in our benchmark. For *extrinsic and focal control*, we compare with **TrajectoryCrafter** [35], which reconstructs a point-cloud representation from the input video via monocular depth estimation and renders novel-view frames under target camera poses, filling occluded regions with a video diffusion prior; and **ReCamMaster** [2], which conditions a video diffusion model on target camera extrinsic matrices using multi-camera synchronized training data rendered with Unreal Engine [6]. For *resolution editing*, we compare with **Follow-Your-Canvas** [4], a video outpainting method that expands the spatial extent of a video while preserving temporal coherence; and **VACE** [17], a general video editing framework used here as a reference-based generation baseline.

*Evaluation metrics.* We evaluate on both synthetic and real-world videos. ***For synthetic evaluation***, we sample 50 unseen scenes from our rendered dataset and construct 6 test cases per scene, yielding 300 test pairs in total. The source video for every test case is a fixed-focal, single-shot recording from the same scene. The six target configurations cover all combinations of extrinsic and intrinsic control dimensions: (1) *source portrait*: the source video re-encoded at portrait resolution (832×480) as a resolution-change reference; (2) *single-shot extrinsics*: a different single-shot camera trajectory at the same focal length; (3) *multi-shot extrinsics*: a multi-shot (cut) trajectory at the same focal length; (4) *focal only*: the source trajectory rendered at a different focal length; (5) *single-shot extrinsics + focal*: a different single-shot trajectory combined with a focal-length change; (6) *multi-shot extrinsics + focal*: a multi-shot trajectory combined with a focal-length change. We report PSNR ↑, SSIM ↑, and LPIPS ↓ [38] to measure reconstruction fidelity against the ground-truth target videos. Perceptual quality is assessed using FVD ↓ [30], and six VBench [15] dimensions: aesthetic quality, imaging quality, temporal flickering, motion smoothness, subject consistency, and background consistency. ***For real-world evaluation***, we select 50 videos from the DAVIS dataset [24] and pair each with 6 synthesized camera trajectories following the same scheme, yielding 300 test pairs. Since no ground-truth target videos exist for these pairs, we evaluate *camera accuracy* and *perceptual quality*. For camera accuracy, we run ViPE [13] on each generated video to estimate per-frame poses and compare against the target trajectory. We report **RotErr** (°, ↓), the mean rotation error after normalising both trajectories to

**Table 1:** Quantitative comparison with state-of-the-art methods on the synthetic dataset. We report three categories of evaluation: Reconstruction (Recon.), VBench [15], and Video Quality (V.Q.). Extrinsic and focal length controllability are evaluated against TrajectoryCrafter (Traj.) [35] and ReCamMaster (ReCam.) [2], while resolution adaptation is compared with Follow-Your-Canvas (F-Y-C) [4] and VACE [17].

| Category | Metric | Ext./Focal | | | Resolution Editing | | |
|---|---|---|---|---|---|---|---|
| | | Traj. | ReCam. | **Ours** | F-Y-C | VACE | **Ours** |
| Recon. | PSNR↑ | 12.24 | 12.87 | **15.88** | 19.91 | 19.45 | **20.54** |
| | SSIM↑ | 0.379 | 0.379 | **0.482** | **0.765** | 0.714 | 0.758 |
| | LPIPS↓ | 0.654 | 0.607 | **0.179** | 0.206 | 0.194 | **0.174** |
| VBench | Aesthetic↑ | 0.4726 | 0.5338 | **0.5340** | 0.5374 | 0.5460 | **0.5462** |
| | Imaging↑ | 0.5807 | 0.6505 | **0.6695** | 0.6346 | 0.6598 | **0.6614** |
| | Flicker↑ | 0.9565 | **0.9717** | 0.9648 | **0.9670** | 0.9652 | 0.9655 |
| | Motion↑ | 0.9845 | **0.9910** | 0.9881 | 0.9892 | 0.9889 | **0.9903** |
| | Subject↑ | 0.8619 | **0.9084** | 0.8912 | 0.8929 | 0.9014 | **0.9030** |
| | Background↑ | 0.9131 | **0.9183** | 0.9071 | 0.9133 | **0.9225** | 0.9100 |
| V.Q. | FVD↓ | 462.9 | 351.5 | **231.0** | 526.8 | 422.7 | **325.8** |

frame 0, and **TransErr** (↓), the mean translation error after scale alignment via arc length. Perceptual quality is evaluated with VBench.

### 5.2 Comparisons with state-of-the-art

*Quantitative results.* Quantitative comparisons are summarized in Tabs. 1 and 2, grouped into *extrinsic/focal control* and *resolution editing* with task-specific baselines. For *extrinsic and focal control*, camera following is the primary objective, and our method follows the target camera more accurately. It improves synthetic reconstruction over both TrajectoryCrafter [35] and ReCamMaster [2] (PSNR: **15.88** vs. 12.24 and 12.87) and achieves lower real-world camera error (RotErr: **2.76**° vs. 5.12° for ReCamMaster and 13.99° for TrajectoryCrafter; TransErr: **0.33** vs. 0.46 and 0.99). ReCamMaster scores higher on several VBench dimensions because it keeps the first frame identical to the source and does not perform multi-shot transitions, thereby preserving more input pixels and synthesizing fewer novel regions. In contrast, our method targets harder novel-view and camera-cut settings, yielding substantially stronger camera accuracy and better Imaging Quality on both benchmarks. In supplementary material, we also provide a user study showing higher human preference for camera alignment and overall video quality.

For *resolution editing*, our method performs consistently well across synthetic and real-world evaluations, with advantages in reconstruction (PSNR/LPIPS), video quality (FVD), and key perceptual metrics such as Aesthetic, Imaging Quality, and Motion Smoothness over Follow-Your-Canvas [4] and VACE [17]. VACE attains slightly higher Subject and Background Consistency because its outpainting pipeline preserves the source crop, while our method re-generates the full frame to support native resolution adaptation. These results show the

**Table 2:** Quantitative comparison on real-world DAVIS videos. We report Camera Accuracy (Cam. Acc.) and VBench [15]. Extrinsic camera control is compared with TrajectoryCrafter (Traj.) [35] and ReCamMaster (ReCam.) [2]. Resolution adaptation is compared with Follow-Your-Canvas (F-Y-C) [4] and VACE [17]. "–" indicates the method does not support this task.

| Category | Metric | Ext./Focal | | | Resolution Editing | | |
|---|---|---|---|---|---|---|---|
| | | Traj. | ReCam. | **Ours** | F-Y-C | VACE | **Ours** |
| Cam. Acc. | RotErr↓ | 13.9916 | 5.11982 | **2.7600** | – | – | – |
| | TransErr↓ | 0.9949 | 0.4628 | **0.3309** | – | – | – |
| VBench | Aesthetic↑ | 0.4291 | **0.5166** | 0.4752 | 0.5166 | 0.5157 | **0.5198** |
| | Imaging↑ | 0.6004 | 0.6833 | **0.6857** | 0.6728 | 0.6644 | **0.6732** |
| | Flicker↑ | 0.9077 | **0.9596** | 0.9463 | 0.9408 | 0.9395 | **0.9413** |
| | Motion↑ | 0.9527 | **0.9884** | 0.9809 | 0.9718 | 0.9743 | **0.9790** |
| | Subject↑ | 0.8142 | **0.9094** | 0.8410 | 0.8898 | **0.8925** | 0.8908 |
| | Background↑ | 0.8857 | **0.9110** | 0.8828 | 0.9125 | **0.9271** | 0.9127 |

**Table 3:** Camera representation and injection ablation. "Linear" denotes a frame-level 21-d encoding of extrinsic and intrinsic matrices; "Plücker Ray" denotes per-pixel 6-DoF ray embeddings [26]. Injection methods are element-wise addition (Add), adaptive layer normalization (AdaLN), and RoPE-based phase shift (RoPE).

| Representation | Injection | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|---|
| Linear | Addition | 15.62 | 0.481 | 0.434 |
| Plücker Ray | AdaLN | 16.51 | 0.517 | 0.368 |
| Plücker Ray | RoPE | **16.66** | **0.528** | **0.372** |

benefit of jointly modeling camera geometry and resolution changes within a unified framework.

*Qualitative results.* Fig. 5 shows real-world comparisons under large viewpoint changes and multi-shot retake trajectories. TrajectoryCrafter often suffers from geometric errors and visual artifacts, while ReCamMaster [2] keeps the first frame identical to the source video and struggles to follow multi-shot camera transitions. Our method follows complex camera controls more accurately while preserving geometric consistency and visual quality. The in-the-wild examples in Fig. 6 further demonstrate its generalization to real scenes despite training only on synthetic data.

### 5.3 Ablation studies and analysis

*Camera representation and injection.* Table 3 compares three representation and injection choices with all other settings fixed. Compared with linear representation and token-wise addition (row 1), row 2 improves all metrics by using Plücker rays for per-pixel geometric cues and AdaLN to modulate token activations through learned scale and shift, instead of directly perturbing their values.

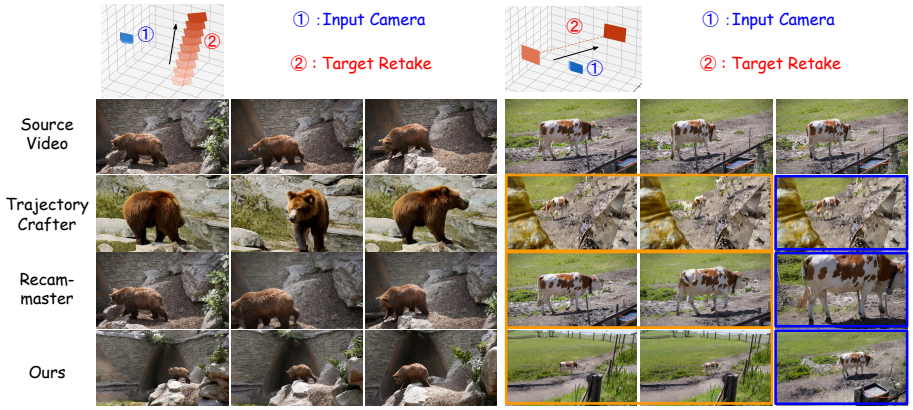


**Fig. 5:** Qualitative comparison on real-world videos with TrajectoryCrafter [35] and ReCamMaster [2]. Given the same source video and target retake camera parameters, our method demonstrates more accurate camera trajectory following under challenging scenarios, including large viewpoint changes and multi-cut shot transitions, while maintaining higher visual fidelity and geometric consistency.


**Fig. 6:** In-the-wild results. Our method generalizes to diverse real-world scenes despite being trained on synthetic data only.

Replacing AdaLN with RoPE injection (row 3) further improves performance, suggesting that injecting camera geometry through attention is more effective than feature-space modulation. We include qualitative comparisons for different injection designs in Fig. 7.

*Training strategy.* Table 4 ablates our joint training strategy with two variants: *extrinsic-only* and *without identity samples*. Extrinsic-only training disables focal and resolution changes, yet performs comparably to the full model on extrinsic tasks, confirming that jointly training focal length and resolution does not degrade extrinsic control. Without identity samples, all control dimensions change together, and performance drops consistently across all three task groups. This shows that identity samples provide the data-driven supervision needed to decouple extrinsic, focal, and resolution controls during joint training.


**Fig. 7:** Visual comparisons for the ablation study. Given the same source video and target retake camera parameters, Plücker ray representation with RoPE-based injection yields stronger geometric consistency, with fewer artifacts (highlighted by red dashed boxes) compared to other variants. Green dashed boxes indicate the ground truth.

**Table 4:** Training strategy ablation on the synthetic benchmark. *Extrinsic-only* disables focal/resolution changes; *W/o identity samples* makes all control dimensions change together. Best in **bold**.

| Configuration | Extrinsic Tasks | Focal Tasks | Resolution Tasks |
|---|---|---|---|
| | PSNR↑ / SSIM↑ / LPIPS↓ | PSNR↑ / SSIM↑ / LPIPS↓ | PSNR↑ / SSIM↑ / LPIPS↓ |
| Full model | **14.13** / **0.423** / **0.483** | **20.03** / **0.606** / **0.213** | **20.05** / **0.743** / **0.187** |
| Extrinsic-only | 14.00 / 0.414 / 0.497 | – | – |
| W/o identity samples | 13.87 / 0.406 / 0.503 | 15.94 / 0.475 / 0.399 | 14.64 / 0.456 / 0.414 |

## 6 Conclusion

We presented CameraAnything, a unified framework for camera-controlled video editing that supports multi-shot retakes, focal-length, and resolution changes in a single generation process. We inject per-pixel Plücker rays into resolution-aware 3D RoPE in self-attention. We further built a scalable synthetic dataset via structured multi-camera recording, with synchronized clip pairs covering extrinsic, focal, and resolution variations. Moreover, we designed an orthogonal training strategy that independently samples extrinsic, focal, and resolution edits, enabling both single-dimension control and their compound combinations. Experiments demonstrate strong benchmark performance and practical promise for cinematic production and cross-platform video adaptation.

## Acknowledgements

This work was supported by Ant Group Research Intern Program and Ant Group Postdoctoral Programme.

## References


1. Adobe: Dolly zoom shot: Definition and examples. https://www.adobe.com/sg/creativecloud/video/production/cinematography/camera-shots-and-angles/dolly-zoom-shot.html (2026), accessed: 2026-06-30
2. Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X., Wan, P., et al.: Recammaster: Camera-controlled generative rendering from a single video. In: ICCV (2025)
3. Bai, J., Xia, M., Wang, X., Yuan, Z., Liu, Z., Hu, H., Wan, P., ZHANG, D.: Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. In: ICLR (2025)
4. Chen, Q., Ma, Y., Wang, H., Yuan, J., Zhao, W., Tian, Q., Wang, H., Min, S., Chen, Q., Liu, W.: Infinite-canvas: Higher-resolution video outpainting with extensive content generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 2150–2158 (2025)
5. Chen, Y., Ye, Z., Fang, Z., Chen, X., Zhang, X., Liu, J., Wang, N., Liu, H., Zhang, G.: Postcam: Camera-controllable novel-view video generation with query-shared cross-attention. arXiv preprint arXiv:2511.17185 (2025)
6. Epic Games: Unreal Engine. https://www.unrealengine.com/ (2023), accessed: 2026-06-30
7. Fu, X., Tang, S., Shi, M., Liu, X., Gu, J., Liu, M.Y., Lin, D., Lin, C.H.: Plenoptic video generation. arXiv preprint arXiv:2601.05239 (2026)
8. Gao, R., Hołyński, A., Henzler, P., Brussee, A., Martin-Brualla, R., Srinivasan, P., Barron, J.T., Poole, B.: Cat3d: create anything in 3d with multi-view diffusion models. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. pp. 75468–75494 (2024)
9. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for text-to-video generation. In: ICLR (2025)
10. He, H., Yang, C., Lin, S., Xu, Y., Wei, M., Gui, L., Zhao, Q., Wetzstein, G., Jiang, L., Li, H.: Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. In: ICCV (2025)
11. Hong, Y., Lee, S., Chung, H., Ye, J.C.: Inversecrafter: Efficient video recapture as a latent domain inverse problem. arXiv preprint arXiv:2512.05672 (2025)
12. Hu, T., Peng, H., Liu, X., Ma, Y.: Ex-4d: Extreme viewpoint 4d video synthesis via depth watertight mesh. arXiv preprint arXiv:2506.05554 (2025)
13. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C., Ren, J., Xie, K., Biswas, J., Leal-Taixé, L., Fidler, S.: Vipe: Video pose engine for 3d geometric perception. CoRR **abs/2508.10934** (2025)
14. Huang, Z., Jeong, H., Chen, X., Gryaditskaya, Y., Wang, T.Y., Lasenby, J., Huang, C.H.: Spacetimepilot: Generative rendering of dynamic scenes across space and time. arXiv preprint arXiv:2512.25075 (2025)
15. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: CVPR (2024)
16. HunyuanWorld, T.: Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. arXiv preprint (2025)
17. Jiang, Z., Han, Z., Mao, C., Zhang, J., Pan, Y., Liu, Y.: Vace: All-in-one video creation and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17191–17202 (2025)

18. Kong, X., Liu, S., Lyu, X., Taher, M., Qi, X., Davison, A.J.: Eschernet: A generative model for scalable view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9503–9513 (2024)
19. Li, R., Yi, B., Liu, J., Gao, H., Ma, Y., Kanazawa, A.: Cameras as relative positional encoding. Advances in Neural Information Processing Systems (2025)
20. Liu, T., Chen, Z., Huang, Z., Xu, S., Zhang, S., Ye, C., Li, B., Cao, Z., Li, W., Zhao, H., et al.: Light-x: Generative 4d video rendering with camera and illumination control. arXiv preprint arXiv:2512.05115 (2025)
21. Luo, Y., Shi, X., Bai, J., Xia, M., Xue, T., Wang, X., Wan, P., Zhang, D., Gai, K.: Camclonemaster: Enabling reference-based camera control for video generation. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–10 (2025)
22. Miyato, T., Jaeger, B., Welling, M., Geiger, A.: Gta: A geometry-aware attention mechanism for multi-view transformers. In: The Twelfth International Conference on Learning Representations (2024)
23. Park, B., Kim, B.H., Chung, H., Ye, J.C.: Redirector: Creating any-length video retakes with rotary camera encoding. arXiv preprint arXiv:2511.19827 (2025)
24. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Gool, L.V.: The 2017 DAVIS challenge on video object segmentation. CoRR **abs/1704.00675** (2017)
25. Seo, J., Han, J., Jung, J., Jin, S., Lee, J., Narihira, T., Fukuda, K., Shibuya, T., Ahn, D., Hu, S., et al.: Vid-camedit: Video camera trajectory editing with generative rendering from estimated geometry. arXiv preprint arXiv:2506.13697 (2025)
26. Sitzmann, V., Rezchikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. Advances in Neural Information Processing Systems **34**, 19313–19325 (2021)
27. Song, C., Yang, Y., Zhao, T., Li, R., Zhang, C.: Worldforge: Unlocking emergent 3d/4d generation in video diffusion model via training-free guidance. arXiv preprint arXiv:2509.15130 (2025)
28. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
29. Sun, H., Shen, L., Peng, Z., Wang, K., Wu, S., Zang, Y., Liu, T., Huang, Z., Zeng, X., Cao, Z., et al.: Generative photographic control for scene-consistent video cinematic editing. arXiv preprint arXiv:2511.12921 (2025)
30. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Fvd: A new metric for video generation. arXiv preprint arXiv:1812.01717 (2019)
31. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025)
32. Wang, Y., Zhang, Q., Cai, S., Wu, T., Ackermann, J., Kuang, Z., Zheng, Y., Rajič, F., Tang, S., Wetzstein, G.: Bullettime: Decoupled control of time and camera pose for video generation. arXiv preprint arXiv:2512.05076 (2025)
33. Wu, X., Chen, X., Wang, Y., Qiao, Y.: Shotdirector: Directorially controllable multi-shot video generation with cinematographic transitions. arXiv preprint arXiv:2512.10286 (2025)
34. Yang, Q., Yang, Y., Zeng, Y., Hu, X., Li, B., Yue, H., Yang, J., Jiang, P.T.: Cameramaster: Unified camera semantic-parameter control for photography retouching. arXiv preprint arXiv:2511.21024 (2025)
35. Yu, M., Hu, W., Xing, J., Shan, Y.: Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 100–111 (October 2025)

36. Yuan, Y., Wang, X., Sheng, Y., Chennuri, P., Zhang, X., Chan, S.: Generative photography: Scene-consistent camera control for realistic text-to-image synthesis. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 7920–7930 (2025)
37. Zhang, C., Li, B., Wei, M., Cao, Y.P., Gambardella, C.C., Phung, D., Cai, J.: Unified camera positional encoding for controlled video generation. arXiv preprint arXiv:2512.07237 (2025)
38. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
39. Zhong, L., Li, F., Huang, Y., Liu, J., Pei, R., Song, F.: Outdreamer: Video outpainting with a diffusion transformer. arXiv preprint arXiv:2506.22298 (2025)