

# Unified Camera Positional Encoding for Controlled Video Generation

Cheng Zhang<sup>1,2</sup> Boying Li<sup>1\*</sup> Meng Wei<sup>1</sup>  
 Yan-Pei Cao<sup>3</sup> Camilo Cruz Gambardella<sup>1,2</sup> Dinh Phung<sup>1</sup> Jianfei Cai<sup>1</sup>  
<sup>1</sup>Monash University <sup>2</sup>Building 4.0 CRC <sup>3</sup>VAST

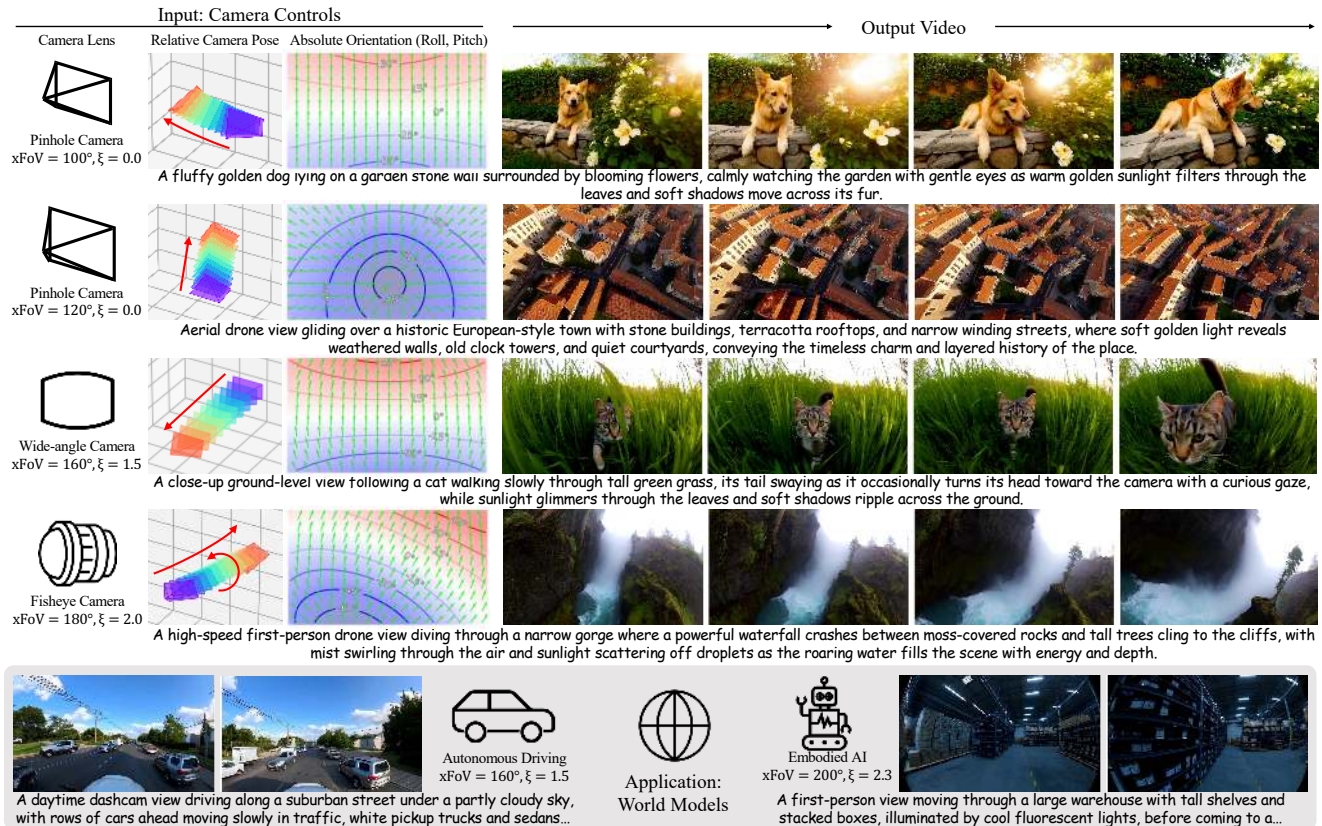


Figure 1. **Overview of our camera-controllable video generation.** Given user-specified text prompt and camera parameters including horizontal field-of-view, distortion  $\xi$ , and camera poses with optional absolute orientation (encoded as latitude-up map), our model synthesizes realistic videos consistent with diverse camera geometries, demonstrating accurate pose and lens controllability and high visual fidelity. Applications span generative video content creation and world models for autonomous driving and embodied AI.

## Abstract

Transformers have emerged as a universal backbone across 3D perception, video generation, and world models for autonomous driving and embodied AI, where understanding camera geometry is essential for grounding visual observations in three-dimensional space. However, existing camera encoding methods often rely on simplified pinhole assump-

tions, restricting generalization across the diverse intrinsics and lens distortions in real-world cameras. We introduce **Relative Ray Encoding**, a geometry-consistent representation that unifies complete camera information, including 6-DoF poses, intrinsics, and lens distortions. To evaluate its capability under diverse controllability demands, we adopt camera-controlled text-to-video generation as a testbed task. Within this setting, we further identify pitch and roll as two components effective for **Absolute Orientation Encoding**, enabling full control over the initial camera

\*Corresponding author.

orientation. Together, these designs form **UCPE (Unified Camera Positional Encoding)**, which integrates into a pre-trained video Diffusion Transformer through a lightweight spatial attention adapter, adding **less than 1% trainable parameters** while achieving state-of-the-art camera controllability and visual fidelity. To facilitate systematic training and evaluation, we construct a large video dataset covering a wide range of camera motions and lens types. Extensive experiments validate the effectiveness of UCPE in camera-controllable video generation and highlight its potential as a general camera representation for Transformers across future multi-view, video, and 3D tasks. Code will be available at <https://github.com/chengzhag/UCPE>.

## 1. Introduction

Transformers [56] have emerged as the foundation of modern architectures for novel view synthesis [27], 3D reconstruction [26], and camera-controllable video generation [1, 11, 19], where networks must reason about how visual observations are formed by camera geometries (e.g., pose, intrinsics, projection model, lens distortion) in order to ground pixel sequences into 3D space. To maximize visual coverage from limited viewpoints, real-world applications in autonomous driving [32, 67], robotic perception [40, 46, 73], and world models [5, 53, 65] routinely use fisheye, catadioptric, and equirectangular projections for 360° panoramas. However, despite progress in Transformer backbones [26, 27, 59, 66], most camera-conditioned methods still rely on simplified pinhole assumptions, overlooking the highly nonlinear projections and strong lens distortions present in these diverse camera geometries.

In practice, these pinhole-based assumptions manifest in the camera representations fed into the network, as summarized in Fig. 2 and detailed in Sec. 3.1, which can be categorized into *absolute* and *relative* encodings. Absolute encodings either feed raw camera parameters directly [3, 62] or adopt Plücker encoding [1, 19] to represent rays under pinhole models. However, as shown in our experiments (Sec. 4.2), their dependence on specific world-frame configurations limits generalization across diverse camera geometries. Recent relative encodings such as GTA [45] and PRoPE [34] improve multi-view consistency in novel view synthesis by injecting geometry- or projection-aware relations into attention. Yet, they remain restricted to the pinhole projection and are not readily compatible with pre-trained video diffusion models. *Consequently, existing camera-conditioning methods still lack an encoding mechanism that explicitly models heterogeneous camera geometries, leading to limited generalization across varying intrinsics, distortion profiles, and viewpoint settings.*

To address these limitations, we propose *Relative Ray Encoding* (Fig. 2d), which reformulates positional encod-

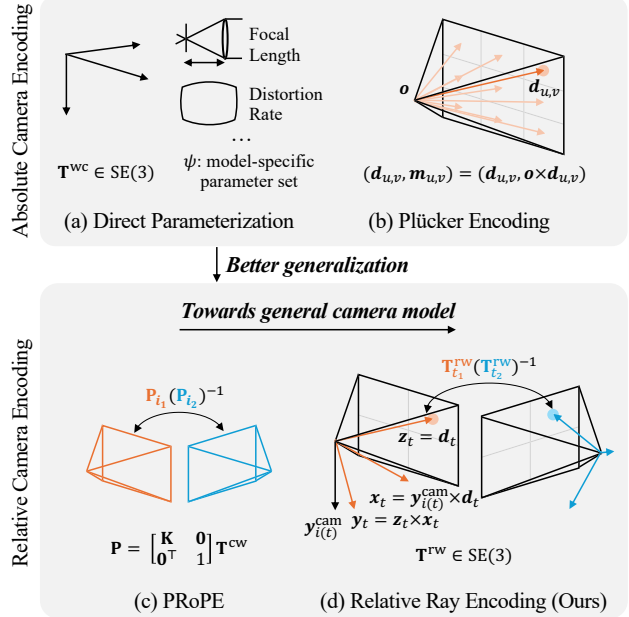


Figure 2. **Comparison of camera encoding methods.** (a) *Direct Parameterization* encodes camera intrinsics and extrinsics as raw parameters, which lacks geometric interpretability and compatibility across camera types. (b) *Plücker Encoding* represents each ray as a pair of direction and moment vectors, providing a physically grounded but absolute, coordinate-dependent description. (c) *Projective Positional Encoding* encodes relative cameras in projective space, yet assumes pinhole projection and cannot model non-linear lens distortions. (d) Our *Relative Ray Encoding* reformulates geometric relationships in ray space, where each token corresponds to its own viewing ray, enabling better pose generalization and compatibility with **arbitrary** camera lenses.

ing from relative-camera encoding into relative-ray encoding. By representing each token as a viewing ray expressed in local ray coordinates, this encoding enables the attention mechanism to operate directly in ray space rather than at the camera level, achieving geometry-consistent conditioning across heterogeneous camera lenses.

To evaluate this capability under diverse controllability demands, we adopt camera-controlled text-to-video generation as a testbed task. However, existing camera-conditioned T2V methods [1, 19] define camera poses only *relative* to the first frame, leaving the global rotation frame ambiguous. In particular, two degrees of freedom—*pitch* and *roll*—cannot be uniquely established, making it impossible to specify or reproduce the absolute orientation of the initial view. To resolve this ambiguity, we introduce an *Absolute Orientation Encoding* (third column of Fig. 1), which anchors the camera to a gravity-aligned “up” direction. This encoding explicitly provides absolute pitch and roll angles as a latitude-up map, enabling precise and repeatable control over the camera’s absolute orientation.

Combining these two components, we arrive at

**UCPE (Unified Camera Positional Encoding)**, a camera model-agnostic framework that injects complete camera geometry—including 6-DoF poses, intrinsics, and distortions—into Transformer attention. UCPE provides a consistent formulation that unifies heterogeneous camera geometries, enabling fine-grained control over lens type, viewpoint, and orientation, as illustrated in Fig. 1. Integrated into diffusion-based video generation through a lightweight spatial attention adapter, UCPE allows existing Diffusion Transformers to be fine-tuned with less than 1% additional parameters.

To support systematic training and evaluation, we further construct a large-scale video dataset spanning diverse intrinsics, distortion profiles, and camera motions, offering a comprehensive benchmark for camera-controllable generation. Experiments demonstrate that UCPE consistently improves camera controllability and visual fidelity across varied camera configurations, establishing a unified bridge between physical camera lenses and attention mechanisms for controllable video generation.

In summary, our contributions are as follows:

- We propose *UCPE*, a unified framework that encodes complete camera geometry (6-DoF poses, intrinsics, lens distortions) into Transformers. UCPE adopts a hybrid formulation combining *Relative Ray Encoding* for ray-space geometric reasoning and projection nonlinearity and *Absolute Orientation Encoding* for a gravity-aligned reference enabling controllable camera orientation.
- We introduce a lightweight *attention injection method* that integrates UCPE into video Diffusion Transformers with less than 1% of additional trainable parameters, enabling camera-aware video generation while preserving the visual fidelity of pretrained models.
- We construct a *large-scale video dataset* covering diverse camera intrinsics, distortion profiles, and motion trajectories, providing a benchmark for evaluating camera-controllable generation under varied camera geometries.
- Extensive experiments demonstrate that UCPE not only enables accurate lens and orientation control, but also improves pose accuracy and generation quality for pinhole cameras and across diverse camera types.

## 2. Related Work

**Video Diffusion Models.** Diffusion-based generative models have become a powerful paradigm for video synthesis, achieving remarkable realism and temporal coherence. Early attempts extended image diffusion frameworks [18, 49] into the video domain by incorporating temporal modules to model motion dynamics. Subsequent UNet-based [7, 23] and Transformer-based [59, 66] architectures operate in 3D video latent spaces to perform spatiotemporal denoising. With the scaling up of training datasets and spatial resolution, large diffusion frameworks, such as Hunyuan-

Video [30] and Seedance [14], have pushed video generation quality to a level nearly indistinguishable from real-world footage. To achieve controllability, these models typically incorporate textual, visual, or image-space conditions [9, 15, 17, 60] (*e.g.*, depth, edges, or optical flow), to guide motion and appearance synthesis. Nevertheless, most pretrained video diffusion frameworks remain agnostic to the underlying camera geometry that fundamentally determine how visual observations are formed.

**Camera-Controlled Generation.** Recent efforts toward camera-controlled video generation aim to synthesize videos under explicit camera control, either through fine-tuning or training-free adaptation. One line of work employs 3D representations to guide novel-view generation using image-space conditions such as rendering [10, 12, 35, 39, 42, 68–70, 74], tracking [15], optical flow [9, 29], or coordinates [72]. These approaches facilitate viewpoint control and improve 3D consistency but depend on estimated depth maps, point clouds, meshes, or 3D Gaussians derived from input images or videos, thereby limiting motion diversity and often degrading under large camera movements or imperfect reconstructions. Another direction conditions diffusion models directly on camera poses for video-to-video [3], image-to-video [13, 20, 36, 75, 76], and text-to-video generation [1, 2, 11, 19, 62], relying on implicit 3D priors learned during training. Although these methods demonstrate promising camera-aware generation, they typically define a reference frame on the first frame, without absolute orientation control (*e.g.*, pitch and roll) in text-to-video synthesis. Moreover, most existing approaches manipulate only extrinsic parameters (*i.e.*, 6-DoF poses) while overlooking intrinsics and lens distortion, which hinders generalization across diverse camera configurations and projection models. *In contrast, our method introduces a unified, geometry-consistent representation that jointly encodes poses, intrinsics, and distortion, enabling physically grounded and accurate camera-aware control across a wide range of projection types.*

**Camera Encoding.** Achieving fine-grained camera control in diffusion models fundamentally depends on how camera geometry is represented and encoded. A few works explore direct camera conditioning by injecting camera parameters into diffusion models [3, 62], enabling controllable viewpoint transitions without explicit 3D reconstruction. Other methods encode camera poses through ray-map encodings for image-to-video [13, 20, 36, 75, 76] and text-to-video generation [1, 2, 19]. Ray-map representations describe each pixel by its corresponding ray origin and direction [13, 44] or by its Plücker coordinates [1, 47, 61], allowing models to incorporate both intrinsic and extrinsic camera properties. However, these absolute encodings rely on a predefined world frame [13, 16, 43], making the representation dependent on arbitrary coordinate choices and limiting

cross-scene generalization.

Motivated by the success of relative positional encodings such as RoPE [21], several studies propose *relative camera encodings* that model pairwise geometric transformations between views [31, 34, 45, 50]. By encoding relative SE(3) transformations directly at the attention level, these approaches remove the need for a fixed reference frame and have been shown to improve multi-view reasoning and novel-view generation. Building on these insights, *our UCPE extends beyond perspective cameras by jointly encoding poses, intrinsics, and distortion within a geometry-consistent formulation*. This unified representation seamlessly integrates with our attention adapter, allowing Diffusion Transformers to reason about camera geometry consistently across different projection types and achieve accurate viewpoint control.

### 3. Method

#### 3.1. Preliminaries

**Camera as Ray Mapping.** All camera models, regardless of their projection types, can be represented under a unified formulation that maps image coordinates to three-dimensional rays in space. Rather than assuming a specific projection model, a camera is defined by a ray-mapping function  $\Phi_\psi : (u, v) \mapsto (\mathbf{o}_{u,v}^{\text{cam}}, \mathbf{d}_{u,v}^{\text{cam}})$ , which produces, for every pixel  $(u, v)$ , a ray origin  $\mathbf{o}_{u,v}^{\text{cam}} \in \mathbb{R}^3$  and a unit direction  $\mathbf{d}_{u,v}^{\text{cam}} \in \mathbb{S}^2$  in the camera coordinate system. The mapping is parameterized by a model-specific set  $\psi$ , e.g., focal length and distortion coefficients, depending on the chosen projection model. For *central* cameras, all rays share a common origin ( $\mathbf{o}_{u,v}^{\text{cam}} = \mathbf{0}$ ), whereas *non-central* cameras, such as catadioptric or panoramic systems, assign pixel-dependent origins. Unless otherwise stated, subsequent derivations assume a central camera model for notational simplicity. We take the Unified Camera Model (UCM) [41] as a representative example (details in Supplementary Sec. A.4).

Let  $\mathbf{T}^{\text{wc}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \text{SE}(3)$  denote the camera pose from the camera to the world frame. Applying this transformation yields the world-space ray representation:

$$\mathbf{d}_{u,v} = \mathbf{R}\mathbf{d}_{u,v}^{\text{cam}}, \quad \mathbf{o}_{u,v} = \mathbf{t}. \quad (1)$$

This provides a common geometric basis for reasoning about cameras with diverse projection characteristics.

**Absolute Camera Encodings** explicitly encode per-camera parameters or per-pixel rays in the world coordinate system. The most direct approach encodes raw camera parameters as numerical feature vectors [3, 62], as shown in Fig. 2a. A more physically grounded alternative is the *Plücker encoding* [1, 19] (Fig. 2b), which reformulates each ray as a six-dimensional vector  $(\mathbf{d}_{u,v}, \mathbf{m}_{u,v}) \in \mathbb{R}^6$ , where

$\mathbf{m}_{u,v} = \mathbf{o}_{u,v} \times \mathbf{d}_{u,v}$  denotes the ray moment. This representation provides a compact and numerically stable description of each pixel’s ray, encoding both camera pose and lens in a physically interpretable form. However, as an absolute encoding, it inherently remains sensitive to the chosen coordinate frame, which limits its generalizability across different scenes and viewpoints.

**Relative Camera Encodings.** Recent work introduces relative encodings to model pairwise camera relationships directly within the attention mechanism of Transformers, making them invariant to global coordinate choices. For each token  $t \in \{1, \dots, T\}$  with its corresponding camera index  $i(t) \in \{1, \dots, N\}$ , a transformation matrix  $\mathbf{D}_t$  is derived from the world-to-camera transformation  $\mathbf{T}_{i(t)}^{\text{cw}} = (\mathbf{T}_{i(t)}^{\text{wc}})^{-1}$  as:

$$\mathbf{D}_t = \mathbf{I}_{d/4} \otimes \mathbf{T}_{i(t)}^{\text{cw}}, \quad (2)$$

where  $\mathbf{I}$  is the identity matrix,  $d$  denotes the feature dimension, and  $\otimes$  is the Kronecker product that replicates the 3D transformation across feature subspaces. The resulting block-diagonal matrix  $\mathbf{D} = \text{blkdiag}(\mathbf{D}_1, \dots, \mathbf{D}_T)$  encodes per-token camera pose, which is applied to token features via token-wise matrix-vector multiplication  $\odot$ , e.g.,  $\mathbf{D} \odot Q = \text{blkdiag}(\mathbf{D}_1 Q_1, \dots, \mathbf{D}_T Q_T)$ .

*Camera Positional Encoding (CaPE)* [31] employs such transformation matrices to inject relative camera poses into self-attention. The attention operation is modified as:

$$O = \text{Attn}(\mathbf{D}^\top \odot Q, \mathbf{D}^{-1} \odot K, V). \quad (3)$$

This replaces each query-key interaction  $Q_{t_1}^\top K_{t_2}$  with  $Q_{t_1}^\top \mathbf{D}_{t_1} \mathbf{D}_{t_2}^{-1} K_{t_2}$ , where  $\mathbf{D}_{t_1} \mathbf{D}_{t_2}^{-1} = \mathbf{I}_{d/4} \otimes \mathbf{T}_{i(t_1)}^{\text{cw}} (\mathbf{T}_{i(t_2)}^{\text{cw}})^{-1}$ , conditioning attention on the relative camera pose.

*Geometric Transform Attention (GTA)* [45] extends this formulation by additionally transforming the value matrix, thereby enforcing geometric consistency during both feature comparison and aggregation:

$$O = \mathbf{D} \odot \text{Attn}(\mathbf{D}^\top \odot Q, \mathbf{D}^{-1} \odot K, \mathbf{D}^{-1} \odot V). \quad (4)$$

Building on this idea, *Projective Positional Encoding (PRoPE)* [34] (Fig. 2c) generalizes the relative transformation by replacing  $\mathbf{T}_i^{\text{cw}}$  with a camera projective matrix  $\mathbf{P}_i = \begin{bmatrix} \mathbf{K}_i & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{T}_i^{\text{cw}}$  under intrinsics  $\mathbf{K}_i$ . This projective formulation captures the complete camera frustum geometry, but is limited to pinhole cameras.

#### 3.2. Unified Camera Positional Encoding

**Relative Ray Encoding.** Existing relative camera encodings assume that all image tokens within the same view share a single, linear projection function. This assumption treats the entire image as one rigid entity, simplifying inter-camera geometric reasoning but ignoring intra-camera deviations in projection geometry. Consequently, such camera-level encodings perform well under the idealized pinhole

model, but struggle to generalize to real cameras exhibiting spatially varying projection behaviors such as lens distortions and wide field-of-view effects.

To accommodate non-linear projection models, we reformulate the encoding from a *camera-to-camera* to a *ray-to-ray* transformation, capturing fine-grained geometric variations across image tokens. Specifically, for each image token  $t$ , we aim to construct a local ray-to-world matrix  $\mathbf{T}_t^{\text{wr}}$ , serving as the geometric operator for attention-level feature transformation. Formally, each image token  $t$  is associated with a viewing ray parameterized by its origin and direction in the world coordinate system:

$$\mathbf{r}_t = (\mathbf{o}_t, \mathbf{d}_t), \quad \mathbf{o}_t \in \mathbb{R}^3, \quad \mathbf{d}_t \in \mathbb{S}^2, \quad (5)$$

where  $(\mathbf{o}_t, \mathbf{d}_t)$  is derived as in Eq. (1). We construct  $\mathbf{T}_t^{\text{wr}}$  through defining a *local ray coordinate system* anchored at the camera center, parameterized by an orthonormal basis  $\mathbf{R}_t^{\text{wr}} = [\mathbf{x}_t, \mathbf{y}_t, \mathbf{z}_t]$ . As shown in Fig. 2d, we take each ray direction  $\mathbf{d}_t$  as the local  $z$ -axis and determine the other two orthogonal axes based on the camera’s downward direction  $\mathbf{y}_{i(t)}^{\text{cam}}$  as follows:

$$\mathbf{z}_t = \mathbf{d}_t, \quad \mathbf{x}_t = \mathbf{y}_{i(t)}^{\text{cam}} \times \mathbf{z}_t, \quad \mathbf{y}_t = \mathbf{z}_t \times \mathbf{x}_t. \quad (6)$$

This orthonormal basis  $\mathbf{R}_t^{\text{wr}}$ , together with the translation,  $\mathbf{t}_t^{\text{wr}} = \mathbf{o}_t$ , forms a ray-to-world transformation:

$$\mathbf{T}_t^{\text{wr}} = \begin{bmatrix} \mathbf{R}_t^{\text{wr}} & \mathbf{t}_t^{\text{wr}} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \quad (7)$$

Each token feature is therefore associated with its specific geometric transform  $\mathbf{T}_t^{\text{rw}} = (\mathbf{T}_t^{\text{wr}})^{-1}$ , mapping world coordinates into the local ray and serving as the geometric operator later for attention-level feature transformation. This formulation enables the attention mechanism to reason in ray space rather than sharing the same positional encoding within a camera frame, providing a consistent and physically interpretable geometric basis for UCPE both within and across frames.

**Absolute Orientation Encoding.** While relative ray encoding enables geometry-consistent reasoning across heterogeneous cameras, it provides no notion of absolute orientation, leaving the first-frame orientation ambiguous. Yet, most real-world videos are captured under a gravity-aligned “up” direction, which naturally defines absolute pitch and roll angles. To incorporate this global orientation, we adopt the *latitude-up map (Lat-Up map)* [6, 28, 57] shown in Fig. 1, which concatenates each ray’s *latitude angle* and corresponding *up-vector*. This representation captures camera rotation through appearance cues such as sky-ground separation and object alignment, while also providing some distortion awareness for wide-angle lenses.

For the *latitude map* calculation, we start from ray  $t$ ’s world-space direction  $\mathbf{d}_t = [d_{t,x}, d_{t,y}, d_{t,z}]^\top$  as defined

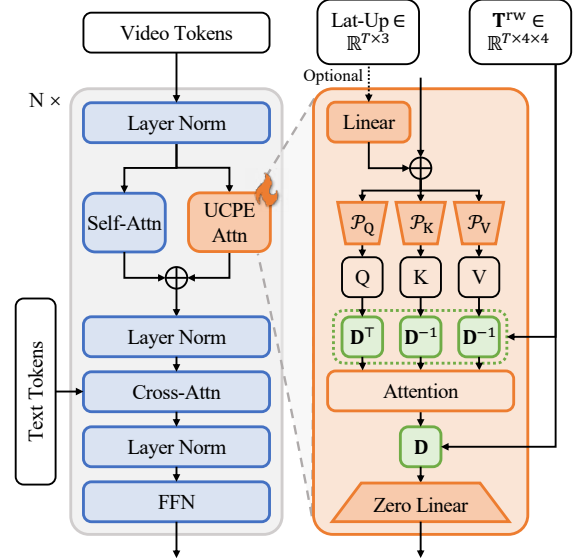


Figure 3. **Overview of Spatial Attention Adapter.** The adapter injects UCPE into pretrained Transformers through a lightweight branch that preserves pretrained priors. It constructs hybrid encoding from the world-to-ray transform  $\mathbf{T}^{\text{rw}}$  and an optional Lat-Up map, applies them within attention, and fuses the resulting camera-aware tokens back through a zero-initialized linear layer.

in Eq. (5). The latitude map then encodes the angular elevation of each ray relative to the horizontal plane:

$$\text{Lat}_t = \arctan 2(-d_{t,y}, \sqrt{d_{t,x}^2 + d_{t,z}^2}), \quad (8)$$

where positive values correspond to upward-looking rays.

To compute the *Up map*, we rotate each world-space ray  $\mathbf{d}_t$  by a small angle  $\delta$  around its local axis  $\mathbf{k}_t = \mathbf{d}_t \times \mathbf{u}^{\text{wld}}$ , where  $\mathbf{u}^{\text{wld}}$  denotes the world up direction. The rotated ray  $\mathbf{d}_t^{\text{rot}}$  is then projected back to the image plane, and the resulting normalized pixel displacement defines the up direction:

$$\text{Up}_t = \frac{[\Delta u_t, \Delta v_t]}{\|[\Delta u_t, \Delta v_t]\|}, \quad (9)$$

where  $[\Delta u_t, \Delta v_t]$  is the projection offset between the rotated and original rays (details in Supplementary Sec. B). The resulting Lat-Up encoding  $[\text{Lat}_t, \text{Up}_t]$  provides each token with a global orientation context, enabling explicit control over camera pitch and roll during generation.

UCPE thus integrates the two complementary cues, *i.e.*, local ray geometry and absolute orientation, to provide consistent and physically interpretable encoding that generalizes across camera lenses.

### 3.3. Spatial Attention Adapter for UCPE Injection

**Spatial Attention with Hybrid Encoding.** We follow [34] to fuse the relative ray encoding with RoPE to promote both ray-space and image-space reasoning. For each token  $t$ , we

construct a block-diagonal operator:

$$\mathbf{D}_t^{\text{UCPE}} = \text{blkdiag}(\mathbf{D}_t^{\text{Ray}}, \mathbf{D}_t^{\text{RoPE}}), \quad (10)$$

where  $\mathbf{D}_t^{\text{Ray}} = \mathbf{I}_{\frac{d}{8}} \otimes \mathbf{T}_t^{\text{rw}} \in \mathbb{R}^{\frac{d}{2} \times \frac{d}{2}}$  encodes the world-to-ray transformation defined in Eq. (7), and  $\mathbf{D}_t^{\text{RoPE}} \in \mathbb{R}^{\frac{d}{2} \times \frac{d}{2}}$  is constructed with RoPE [21] to encode image-space positions, both with half of the feature dimension. We share this design across GTA and PRoPE baselines, which is their best-performing configuration [34]. We then define  $\mathbf{D}^{\text{UCPE}} = \text{blkdiag}(\mathbf{D}_1^{\text{UCPE}}, \dots, \mathbf{D}_T^{\text{UCPE}})$ , and apply it to the attention operation as shown in Fig. 3 following Eq. (4).

Additionally, the Spatial Attention module takes a Lat-Up map as an optional input to provide absolute orientation control. When present, the Lat-Up feature is projected by a linear layer to match the token dimension and then added to the input tokens as bias term.

**Adapting Transformer with Attention.** We adopt Wan [59] as our pretrained Diffusion Transformer, and intuitively the most suitable place to inject UCPE is within its self-attention layers that model spatial and temporal correlations. However, directly replacing the existing positional encoding (*e.g.*, 3D RoPE) with UCPE may disturb the priors established during large-scale pretraining.

To preserve priors, we integrate UCPE through a lightweight, LoRA-inspired adapter that operates in parallel with the original attention. As in Fig. 3, each DiT block retains standard self-attention while introducing a camera-conditioned branch  $\text{UCPEAttn}(\cdot)$  based on the proposed encoding. Crucially, thanks to the geometry-aware design of UCPE, this adapter requires only a small number of parameters to effectively model camera-conditioned correlations. In practice, we linearly project input tokens to  $1/C$  of the original dimension using  $\mathcal{P}_Q, \mathcal{P}_K, \mathcal{P}_V$ , and proportionally reduce the number of attention heads, effectively reducing parameters and computation. The  $\text{UCPEAttn}(\cdot)$  output is then mapped back through a linear projection layer with zero-initialized weights, ensuring that the pretrained model remains unaltered at initialization.

The spatial adapter enables efficient integration of UCPE into Diffusion Transformers, providing fine-grained camera control with minimal additional parameters.

## 4. Experiment

### 4.1. Experimental Setup

**Datasets.** To train and evaluate our method across diverse camera conditions (pose, intrinsics, and distortion) we synthesize a large-scale dataset of  $\sim 48\text{k}$  clips from in-the-wild  $360^\circ$  videos [58] using Unified Camera Model (UCM) [41] with randomized xFoV and distortion  $\xi$ , covering pinhole, wide-angle, fisheye configurations (see Supplementary Sec. A). For out-of-distribution generalization, we additionally evaluate on 100 randomly selected clips from the

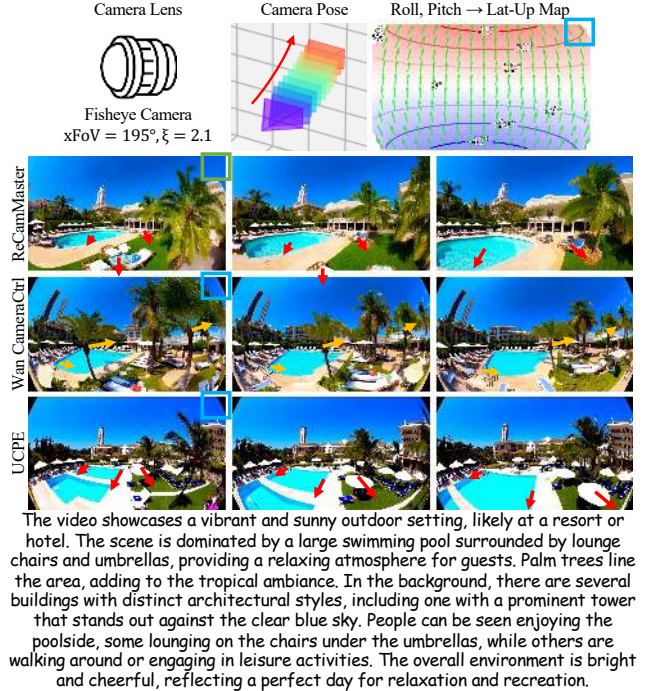


Figure 4. **Comparison on our synthesized dataset.** UCPE faithfully follows target trajectories and produces consistent lens distortions aligned with visualization of the Lat-Up Map. In contrast, Wan CameraCtrl shows camera motion deviations, while ReCamMaster fails to reproduce the intended distortion. Colors correspond to the highlighted effects in the figure.

RealEstate10K test set [77], using a fixed  $100^\circ$  xFoV pinhole camera without any further adaptation or fine-tuning.

**Evaluation Metrics.** We evaluate generated videos in four aspects (detailed in Supplementary Sec. C):

- *Video quality:* evaluated using FID [22], FVD [55] for dataset-level fidelity, CLIP score [48] for text-video alignment and Q-Align [63] for user-centric quality.
- *Relative camera pose control:* each generated frame is rectified to a pinhole view with ground-truth distortion and estimated by ViPE [24], reporting translation (TransErr), rotation (RotErr), and motion consistency (CamMC) [19, 62].
- *Absolute camera orientation:* pitch and roll are estimated via GeoCalib [57] and compared with the ground-truth video estimation to obtain absolute orientation errors.
- *Lens control:* we calibrate FoV and distortion coefficients  $k_1, k_2$  of the radial model [8] with GeoCalib [57] and compute absolute errors against ground-truth calibration.

### 4.2. Comparison with Previous Methods

**Baselines.** We compare UCPE against several representative camera-conditioning methods:

- (1) *ReCamMaster* [3] injects camera extrinsics  $[\mathbf{R}, \mathbf{t}] \in \text{SE}(3)$  into the diffusion model; we extend it by concatenating normalized FoV and distortion  $\xi$  for lens control.
- (2) *Wan CameraCtrl* (Wan2.1-Fun-V1.1-1.3B-Control), a

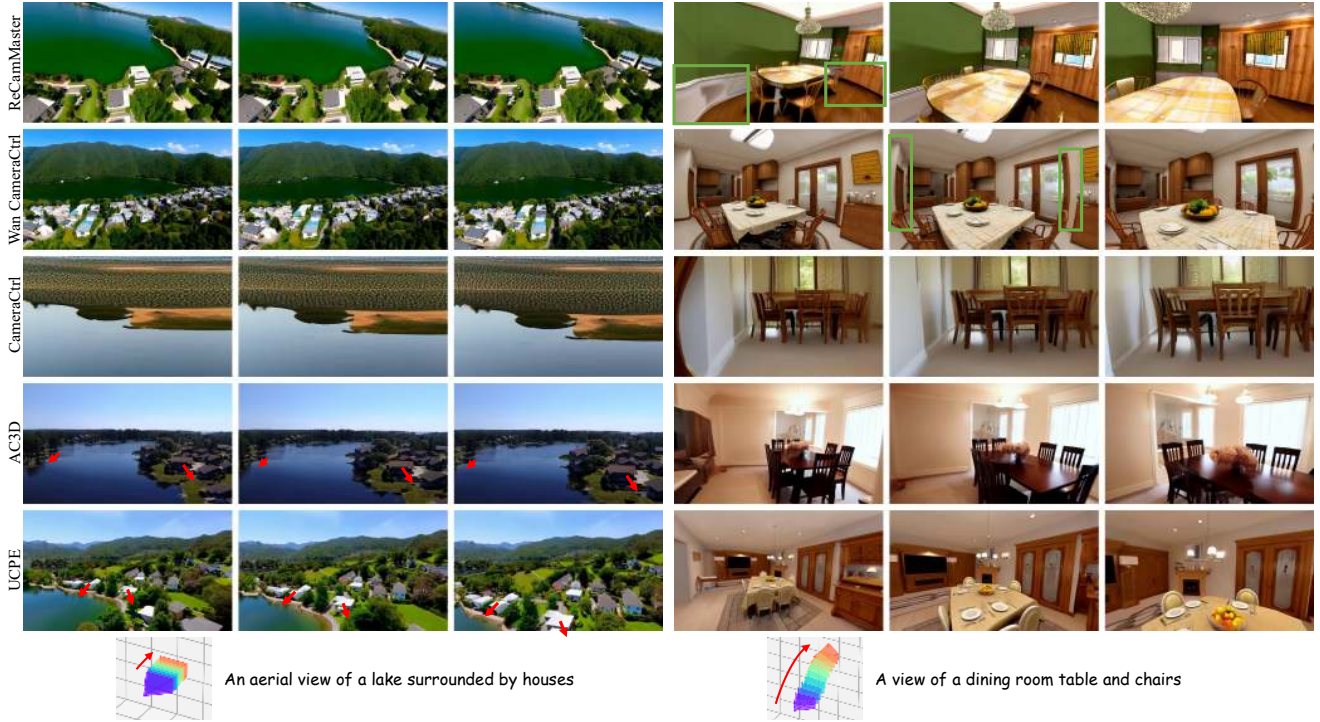


Figure 5. **Comparison on the RealEstate10K dataset.** UCPE generates sharper, more detailed frames that better follow the target camera motion. CameraCtrl produces severe artifacts (left) and poor composition (right), while AC3D preserves the training dataset’s aesthetic but shows unbalanced framing (left) and low dynamic range (right). Wan CameraCtrl and ReCamMaster, though based the same backbone, struggle with camera consistency, leading to reduced motion (left) and **undesired distortion artifacts** (right) under the pinhole setup.

third-party implementation of CameraCtrl [19] with Wan [59], which incorporates Plücker encoding via a convolutional adapter; we adapt it for text-to-video generation. (3) *AC3D* [1] applies Plücker encoding through a ControlNet [71] adapter built on CogVideoX [66].

(4) *CameraCtrl* [19] integrates Plücker encoding into U-Net’s temporal attention layers of AnimateDiff [18].

For both methods (1) and (2), we implement two variants respectively: one following their original design where camera poses are defined relative to the first frame, and another adopting a gravity-aligned coordinate system anchored at the first frame to enable explicit roll and pitch control (denoted as w/ absolute orientation). Methods (1) and (2) share the same base model as UCPE and are fine-tuned on our dataset under identical settings for fair comparison, except that (2) uses a smaller learning rate of  $1e-5$  due to full-parameter optimization (see Supplementary Sec. D for details). For (3) and (4), we directly evaluate the pretrained models released by the authors, and therefore report results only on their training dataset, RealEstate10K [77].

**Quantitative Results.** We present quantitative comparisons on our synthesized dataset in Table 1 and on RealEstate10K in Table 2. On our dataset, we first compare with ReCamMaster and Wan CameraCtrl under their original settings (w/o Absolute Orientation Control), and then under our proposed gravity-aligned coordinate system (w/ Absolute Orientation Control). UCPE consistently achieves bet-

ter performance across both configurations, exhibiting superior camera controllability and video quality while adding only 35.5M or 0.5% parameters over the 7.3B parameters of the base model, 90% fewer than ReCamMaster. The Lat-Up map also provides appearance cues that improve lens control (UCPE w/ vs. w/o Absolute Orientation).

On RealEstate10K, despite not being fine-tuned on this dataset, UCPE outperforms all baselines in Relative Camera Pose Control, demonstrating strong generalization to unseen trajectories and text prompts. Note that UCPE is trained with detailed scene descriptions (see Fig. 4) whereas RealEstate10K uses short prompts (see Fig. 5). Nevertheless, UCPE achieves higher Q-Align scores than CameraCtrl and AC3D, indicating better video quality, and comparable performance to ReCamMaster and Wan CameraCtrl with fewer parameters.

**Qualitative Results.** We visualize generated samples from our synthesized dataset in Fig. 4 and from RealEstate10K in Fig. 5. On our dataset, UCPE **faithfully follows the specified camera trajectories** and produces **consistent lens distortion effects** aligned with the target camera parameters visualized in the Lat-Up Map. In contrast, Wan CameraCtrl exhibits noticeable **camera motion deviations**, while ReCamMaster **fails to reproduce the intended lens distortion**. On RealEstate10K, UCPE generates sharper and more detailed videos that better adhere to the camera motion, whereas CameraCtrl shows severe artifacts (left) and poor

Table 1. **Quantitative comparison on our synthesized dataset.** UCPE outperforms all baselines across lens, orientation, and pose control while maintaining strong video quality with 90% fewer parameters than ReCamMaster. Under both w/o and w/ Absolute Orientation Control, it yields lower pitch, roll, and rotation errors. Ablation results (bottom) show that moderate compression ratios in the Spatial Attention Adapter (e.g., 1/8-dim) best balance controllability and fidelity. Gray cells denote metrics not applicable due to missing control. Here,  $1/C$ -dim denotes the token projection compression ratio, and  $(d \times n)$  are the per-head dimension and number of attention heads.

Settings	Method	Camera Lens Control			Absolute Orientation		Relative Camera Pose Control			Video Generation Quality			Trainable Params
		FoV ( $^\circ$ ) $\downarrow$	$k_1$ $\downarrow$	$k_2$ $\downarrow$	Pitch ( $^\circ$ ) $\downarrow$	Roll ( $^\circ$ ) $\downarrow$	RotErr ( $^\circ$ ) $\downarrow$	TransErr $\downarrow$	CamMC $\downarrow$	FVD $\downarrow$	FID $\downarrow$	CLIP $\uparrow$	
w/o Absolute Orientation Control	ReCamMaster	10.25	0.210	0.143	10.00	7.42	10.89	31.44	37.38	555.54	69.91	24.86	354M
	Wan CameraCtrl	10.05	0.222	0.142	9.35	7.31	17.04	35.09	46.10	593.10	67.83	25.05	1.5B
	UCPE	9.62	0.174	0.120	8.03	6.64	4.29	13.46	15.94	569.31	66.22	25.11	35.5M
w/ Absolute Orientation Control	ReCamMaster	10.04	0.183	0.136	6.62	5.29	9.23	28.95	33.88	605.83	67.07	24.84	354M
	Wan CameraCtrl	9.86	0.230	0.162	6.25	6.01	17.92	39.16	50.32	554.43	65.73	25.01	1.5B
	UCPE	8.22	0.129	0.102	4.35	3.74	4.12	15.21	17.59	495.14	63.37	25.12	35.6M
w/ Absolute Orientation Control & Ablation Study	1/2-dim (128 $\times$ 6)	8.39	0.170	0.110	4.11	3.93	3.69	14.03	16.06	534.44	64.88	25.05	141M
	1/4-dim (128 $\times$ 3)	8.47	0.149	0.101	3.94	4.08	3.43	14.26	16.02	512.85	62.86	25.09	71.0M
	1/8-dim (192 $\times$ 1)	8.22	0.129	0.102	4.35	3.74	4.12	15.21	17.59	495.14	63.37	25.12	35.6M
	1/12-dim (128 $\times$ 1)	8.96	0.151	0.107	3.91	3.89	5.13	14.54	17.84	487.54	62.98	25.06	23.8M
	Pre-Attn	8.47	0.145	0.108	4.26	3.96	4.03	15.77	17.85	502.73	63.62	25.07	35.6M
	Post-Attn	8.91	0.147	0.116	3.95	3.92	4.68	17.47	20.00	515.32	64.65	25.00	35.6M
	PRoPE	8.84	0.151	0.113	4.18	3.70	5.35	17.52	20.58	516.59	65.00	25.03	35.6M
GTA	8.80	0.157	0.117	4.21	3.69	5.27	17.07	20.14	497.19	64.91	25.04	35.6M	

Table 2. **Quantitative comparison on RealEstate10K.** UCPE generalizes well without fine-tuning, achieving the lowest rotation, translation, and motion errors, and showing higher Q-Align scores than models trained on RealEstate10K (CameraCtrl and AC3D).

Method	Relative Camera Pose Control			Q-Align Scores		
	RotErr ( $^\circ$ ) $\downarrow$	TransErr $\downarrow$	CamMC $\downarrow$	Image Quality $\uparrow$	Image Aesthetic $\uparrow$	Video Quality $\uparrow$
ReCamMaster	1.10	5.64	6.15	0.9492	0.5185	0.9720
Wan CameraCtrl	2.22	7.42	8.67	0.9822	0.5691	0.9885
CameraCtrl	1.17	3.96	4.59	0.6877	0.3306	0.7338
AC3D	0.62	2.11	2.43	0.7699	0.3651	0.8211
UCPE	0.56	1.25	1.58	0.9480	0.4686	0.9694

frame composition (right). AC3D maintains the overall aesthetic of RealEstate10K due to training on this dataset but suffers from unbalanced framing (left) and low dynamic range (overexposed window, right). Although Wan CameraCtrl and ReCamMaster are fine-tuned on the same base model and dataset as ours, they struggle to ensure camera consistency, resulting in reduced camera motion (left) and **undesired distortion artifacts** (right) under the pinhole setup.

### 4.3. Ablation Study

We conduct ablation experiments on our synthesized dataset to analyze the core designs of UCPE and the Spatial Attention Adapter, with results summarized in Table 1 (bottom).

**Compression Ratio of Token Projection.** As introduced in Sec. 3.3, we apply a token projection with compression ratio  $1/C$  in the Spatial Attention Adapter to balance efficiency and representation capacity. We test four ratios, each corresponding to different per-head dimensions and numbers of attention heads. Higher compression ratios improve video quality metrics while maintaining stable camera controllability. We adopt 1/8 as the default setting, which provides the best trade-off between performance and efficiency.

**Injection Position of the Attention Adapter.** We compare three configurations within the Diffusion Transformer: insertion before self-attention (Pre-Attn), after self-attention

(Post-Attn), and our default parallel design (1/8-dim). Both Pre-Attn and Post-Attn variants yield noticeably degraded results in both camera control and video quality, confirming the effectiveness of our parallel architecture.

**Comparison with Other Relative Camera Encodings.** To assess the contribution of our relative ray encoding, we replace it with prior formulations including PRoPE [34] and GTA [45], while retaining the Lat-Up Map for orientation and lens cues. Compared to UCPE (1/8-dim), both alternatives lead to weaker camera lens and relative pose control, as well as lower video quality, though they achieve similar absolute orientation performance via the Lat-Up Map. We attribute this to their limited capacity to model non-linear lens projections: both methods use a single camera encoding for all tokens, lacking the ability to capture distortion variation. This limitation is especially pronounced for PRoPE, which even underperforms GTA on several metrics, highlighting the advantage of UCPE’s relative ray encoding as a unified representation across diverse camera lens.

## 5. Conclusion

We presented UCPE, which jointly models poses, intrinsics, and lens distortions through relative ray encoding and absolute orientation cues. Integrated via a lightweight spatial attention adapter, it enables video Diffusion Transformers to achieve accurate camera control with under 1% additional parameters. Experiments on our camera-diverse dataset show clear gains in lens controllability, orientation accuracy, and pose fidelity. Overall, UCPE provides a unified representation for diverse cameras, potentially as a general encoding across multi-view, video and 3D tasks.

**Acknowledgement:** This research is supported by Building 4.0 CRC and partially supported by the Australian Research Council under Grant DP260100218.

## References

- [1] Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *CVPR*, pages 22875–22889, 2025. [2](#), [3](#), [4](#), [7](#)
- [2] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. Vd3d: Taming large video diffusion transformers for 3d camera control. In *ICLR*, 2025. [3](#)
- [3] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *ICCV*, 2025. [2](#), [3](#), [4](#), [6](#), [8](#)
- [4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. [2](#), [6](#)
- [5] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *CVPR*, pages 15791–15801, 2025. [2](#)
- [6] Edurne Bernal-Berdun, Ana Serrano, Belen Masia, Matheus Gadelha, Yannick Hold-Geoffroy, Xin Sun, and Diego Gutierrez. Precisecam: Precise camera control for text-to-image generation. In *CVPR*, pages 2724–2733, 2025. [5](#)
- [7] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. [3](#)
- [8] Dean Brown. Decentering distortion of lenses. 1966. [6](#), [8](#)
- [9] Ryan Burgert, Yuanheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *CVPR*, pages 13–23, 2025. [3](#)
- [10] Chenjie Cao, Jingkai Zhou, shikai Li, Jingyun Liang, Chao-hui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3c: Unifying precisely 3d-enhanced camera and human motion controls for video generation. *ACM SIGGRAPH Conf. Papers*, 2025. [3](#)
- [11] Soon Yau Cheong, Duygu Ceylan, Armin Mustafa, Andrew Gilbert, and Chun-Hao Paul Huang. Boosting camera motion control for video diffusion transformers. *arXiv preprint arXiv:2410.10802*, 2024. [2](#), [3](#)
- [12] Wanquan Feng, Jiawei Liu, Pengqi Tu, Tianhao Qi, Mingzhen Sun, Tianxiang Ma, Songtao Zhao, Siyu Zhou, and Qian He. I2vcontrol-camera: Precise video camera control with adjustable motion strength. In *ICLR*, 2025. [3](#)
- [13] Ruiqi Gao, Aleksander Holyński, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: create anything in 3d with multi-view diffusion models. In *NeurIPS*, pages 75468–75494, 2024. [3](#)
- [14] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025. [3](#)
- [15] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *ACM SIGGRAPH Conf. Papers*, pages 1–12, 2025. [3](#)
- [16] Vitor Guizilini, Muhammad Zubair Irshad, Dian Chen, Greg Shakhnarovich, and Rares Ambrus. Zero-shot novel view and depth synthesis with multi-view geometric diffusion. *arXiv preprint arXiv:2501.18804*, 2025. [3](#)
- [17] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. In *ECCV*, pages 330–348. Springer, 2024. [3](#)
- [18] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024. [3](#), [7](#)
- [19] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024. [2](#), [3](#), [4](#), [6](#), [7](#), [8](#), [9](#)
- [20] Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. *arXiv preprint arXiv:2503.10592*, 2025. [3](#)
- [21] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoon Yun. Rotary position embedding for vision transformer. In *ECCV*, pages 289–305. Springer, 2024. [4](#), [6](#)
- [22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. [6](#)
- [23] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. pages 8633–8646, 2022. [3](#)
- [24] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, et al. Vipe: Video pose engine for 3d geometric perception. *arXiv preprint arXiv:2508.10934*, 2025. [6](#), [2](#), [7](#)
- [25] Breakthrough Apps Inc. and contributors. Pyscenedetect, 2024. [1](#)
- [26] Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. *arXiv preprint arXiv:2505.00702*, 2025. [2](#)
- [27] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *ICLR*, 2025. [2](#)
- [28] Linyi Jin, Jianming Zhang, Yannick Hold-Geoffroy, Oliver Wang, Kevin Blackburn-Matzen, Matthew Sticha, and

- David F Fouhey. Perspective fields for single image camera calibration. In *CVPR*, pages 17307–17316, 2023. 5
- [29] Wonjoon Jin, Qi Dai, Chong Luo, Seung-Hwan Baek, and Sunghyun Cho. Flovd: Optical flow meets video diffusion model for enhanced camera-controlled video synthesis. In *CVPR*, pages 2040–2049, 2025. 3
- [30] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 3
- [31] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschnet: A generative model for scalable view synthesis. In *CVPR*, pages 9503–9513, 2024. 4
- [32] Varun Ravi Kumar, Sandesh Athni Hiremath, Markus Bach, Stefan Milz, Christian Witt, Clément Pinard, Senthil Yogamani, and Patrick Mäder. Fisheyedistancenet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving. In *ICRA*, pages 574–581. IEEE, 2020. 2
- [33] LAION-AI. Laion-5b-watermarkdetection. <https://github.com/LAION-AI/LAION-5B-WatermarkDetection>, 2022. 2
- [34] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. *arXiv preprint arXiv:2507.10496*, 2025. 2, 4, 5, 6, 8
- [35] Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, Chuanyun Deng, Yegan Xiong, Min Chen, et al. Realcam-i2v: Real-world image-to-video generation with interactive complex camera control. In *ICCV*, pages 28785–28796, 2025. 3
- [36] Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N Platanotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. In *CVPR*, pages 798–810, 2025. 3
- [37] Zhiqiu Lin, Siyuan Cen, Daniel Jiang, Jay Karhade, Hewei Wang, Chancharik Mitra, Tiffany Ling, Yuhan Huang, Sifan Liu, Mingyu Chen, et al. Towards understanding camera motions in any video. In *NeurIPS*, 2025. 2
- [38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 8
- [39] Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3d creation on pose-free videos at scale. In *CVPR*, pages 2016–2029, 2025. 3
- [40] Christopher Mei and Patrick Rives. Calibration between a central catadioptric camera and a laser range finder for robotic applications. In *ICRA*, pages 532–537. IEEE, 2006. 2
- [41] Christopher Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. In *ICRA*, pages 3945–3950. IEEE, 2007. 4, 6, 5
- [42] YOU Meng, Zhiyu Zhu, LIU Hui, and Junhui Hou. Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. In *ICLR*, 2025. 3
- [43] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM TOG*, 38(4):1–14, 2019. 3
- [44] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3
- [45] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. Gta: A geometry-aware attention mechanism for multi-view transformers. In *ICLR*, 2024. 2, 4, 8
- [46] Shree K Nayar. Catadioptric omnidirectional camera. In *CVPR*, pages 482–488. IEEE, 1997. 2
- [47] Julius Plucker. Xvii. on a new geometry of space. *Philosophical Transactions of the Royal Society of London*, (155): 725–791, 1865. 3
- [48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. Pmlr, 2021. 6
- [49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 3
- [50] Aleksandr Safin, Daniel Duckworth, and Mehdi SM Sajjadi. Repast: Relative pose attention scene representation transformer. *arXiv preprint arXiv:2304.00947*, 2023. 4
- [51] Hao Shi, Yifan Zhou, Kailun Yang, Xiaoting Yin, Ze Wang, Yaozu Ye, Zhe Yin, Shi Meng, Peng Li, and Kaiwei Wang. Panoflow: Learning 360 optical flow for surrounding temporal understanding. *T-ITS*, 24(5):5570–5585, 2023. 2
- [52] Shinya Sumikura, Mikiya Shibuya, and Ken Sakurada. Openvslam: A versatile visual slam framework. In *ACM MM*, pages 2292–2295, 2019. 1
- [53] HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, et al. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. *arXiv preprint arXiv:2507.21809*, 2025. 2
- [54] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. 13(4):376–380, 2002. 2
- [55] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 6
- [56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 2
- [57] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. Geocalib: Learning single-image calibration with geometric optimization. In *ECCV*, pages 1–20. Springer, 2024. 5, 6, 2, 8

- [58] Matthew Wallingford, Anand Bhattad, Aditya Kusupati, Vivek Ramanujan, Matt Deitke, Aniruddha Kembhavi, Roozbeh Mottaghi, Wei-Chiu Ma, and Ali Farhadi. From an image to a scene: Learning to imagine the world from a million 360 videos. *NeurIPS*, 37:17743–17760, 2024. 6, 1
- [59] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2, 3, 6, 7, 8, 9
- [60] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Junyi Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. In *NeurIPS*, pages 7594–7611, 2023. 3
- [61] Xi Wang, Robin Courant, Marc Christie, and Vicky Kalogeiton. Akira: Augmentation kit on rays for optical video generation. In *CVPR*, pages 2609–2619, 2025. 3
- [62] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH Conf. Papers*, pages 1–11, 2024. 2, 3, 4, 6, 8
- [63] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: teaching Imms for visual scoring via discrete text-defined levels. In *ICML*, pages 54015–54029, 2024. 6, 2
- [64] Yifei Xia, Shuchen Weng, Siqi Yang, Jingqi Liu, Chengxuan Zhu, Minggui Teng, Zijian Jia, Han Jiang, and Boxin Shi. Panowan: Lifting diffusion video generation models to 360  $\{\deg\}$  with latitude/longitude-aware mechanisms. *arXiv preprint arXiv:2505.22016*, 2025. 2
- [65] Zhongqi Yang, Wenhong Ge, Yuqi Li, Jiaqi Chen, Haoyuan Li, Mengyin An, Fei Kang, Hua Xue, Baixin Xu, Yuyang Yin, et al. Matrix-3d: Omnidirectional explorable 3d world generation. *arXiv preprint arXiv:2508.08086*, 2025. 2
- [66] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025. 2, 3, 7
- [67] Senthil Yogamani, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O’Dea, Michal Uricár, Stefan Milz, Martin Simon, Karl Amende, et al. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *ICCV*, pages 9308–9318, 2019. 2
- [68] Mark YU, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *ICCV*, 2025. 3
- [69] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE TPAMI*, 2025.
- [70] Shangjin Zhai, Zhichao Ye, Jialin Liu, Weijian Xie, Jiaqi Hu, Zhen Peng, Hua Xue, Danpeng Chen, Xiaomeng Wang, Lei Yang, et al. Stargen: A spatiotemporal autoregression framework with video diffusion model for scalable and controllable scene generation. In *CVPR*, pages 26822–26833, 2025. 3
- [71] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 7
- [72] Qihang Zhang, Shuangfei Zhai, Miguel Angel Bautista Martin, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. World-consistent video diffusion with explicit 3d modeling. In *CVPR*, pages 21685–21695, 2025. 3
- [73] Zichao Zhang, Henri Rebecq, Christian Forster, and Davide Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *ICRA*, pages 801–808. IEEE, 2016. 2
- [74] Zhiyuan Zhang, Dongdong Chen, and Jing Liao. I2v3d: Controllable image-to-video generation with 3d guidance. *arXiv preprint arXiv:2503.09733*, 2025. 3
- [75] Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. Cami2v: Camera-controlled image-to-video diffusion model. *arXiv preprint arXiv:2410.15957*, 2024. 3
- [76] Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. 3
- [77] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM TOG*, 37(4):1–12, 2018. 6, 7

# Unified Camera Positional Encoding for Controlled Video Generation

## Supplementary Material

This supplementary material is organized as follows:

- Sec. A provides additional details on our construction of the camera-diverse dataset.
- Sec. B elaborates the derivation of the absolute orientation encoding.
- Sec. C summarizes the evaluation metrics used in our experiments.
- Sec. D presents implementation details for both our pipeline and the baselines.
- Sec. E discusses the robustness, generalization capability, limitations, and future work.
- Sec. F includes a detailed demo video.

### A. Dataset Construction

The main paper introduced UCPE, a unified camera encoding that jointly models 6-DoF poses, intrinsics, and lens distortions. Training and evaluating such a representation in a camera-controllable video generation setting requires large-scale data with diverse motions, FoVs, and distortions. However, collecting such real-world data is extremely expensive due to the need for multiple lenses, hardware configurations, and extensive manual effort.

To overcome these limitations, we synthesize camera-diverse videos by using 360° panoramic footage as a flexible exploration space, from which videos can be projected into arbitrary virtual cameras.

We design a *three-stage data synthesis pipeline* comprising: (1) *360° panoramic video curation* to extract high-quality, gravity-aligned exploration clips; (2) *realistic rotation emulation* by transferring camera motions from perspective videos with similar translations; and (3) *camera-diverse video synthesis* under varied intrinsics, distortions, and augmented rotations. While Sec. 4.1 of the main paper provides an overview, here we present the full details.

#### A.1. Coordinate Conventions

Throughout this supplementary material, we adopt a consistent 3D coordinate convention for all projection, ray-mapping, and pose-related computations. The coordinate frame is defined as follows:

- the  $x$ -axis points to the right of the image;
- the  $y$ -axis points downward (following image coordinates);
- the  $z$ -axis points forward, aligned with the viewing direction of the ERP camera.

Under this convention, the world-up direction is

$$\mathbf{u}^{\text{wd}} = [0, -1, 0]^{\top}, \quad (\text{A.1})$$

which is used consistently across ERP projections, UCM ray mappings (Sec. A), and the construction of the Up map (Sec. B).

#### A.2. 360° Panoramic Video Curation

A key benefit of UCPE is controllability over absolute pitch and yaw, resolving the ambiguity in conventional camera-controlled T2V generation. This requires all training videos to share a gravity-aware world frame, which is difficult to obtain from in-the-wild footage.

Fortunately, many commercial 360° cameras produce stabilized, gravity-aligned equirectangular panoramas using calibrated lenses and onboard IMUs. This ensures a consistent up vector, enabling virtual cameras defined inside the panoramic sphere to inherit absolute pitch and yaw.

We begin from the large-scale 4K resolution 360° video corpus of [58], containing 24.1k YouTube videos with diverse scenes and motions. These raw videos contain issues such as scene cuts, watermarks, low-quality footage, non-equirectangular projections, or imperfect stabilization. We therefore construct a multi-stage pipeline to filter low-quality clips and obtain high-quality gravity-aligned panoramas suitable for camera-diverse video synthesis.

**Clip Segmentation.** Video generation models require temporally consistent clips. We first apply a threshold-based transition detector [25] to remove hard cuts and black fades, but this method often misses subtle transitions (*e.g.*, cross fades). We then run panoramic visual SLAM [52] on each preliminary segment. Empirically, SLAM fails to track features across scene transitions, leading to “tracking-lost” signals. By monitoring these signals, we split the preliminary clips into more refined segments with consistent scenes, which results in roughly 400k clips, each of 10 seconds.

**Camera Pose and Rotation Score Extraction.** We run second pass of SLAM in localization mode to obtain more accurate poses, yielding approximately 300k clips with valid estimates. As discussed earlier, enabling absolute orientation supervision requires removing clips that are unstabilized or not gravity-aligned. Since stabilized 360° videos are typically gravity-consistent, we examine the rotational components of the estimated poses and discard clips exhibiting large drift. To quantify this, we compute a rotation score defined as the maximum relative rotation between

each frame and the first frame:

$$\alpha_{\max} = \max_{i>0} \arccos \left( \frac{\text{Tr}(\mathbf{R}_0^\top \mathbf{R}_i) - 1}{2} \right), \quad (\text{A.2})$$

where  $\mathbf{R}_0$  is the first-frame rotation and  $\mathbf{R}_i$  denotes the rotation at frame  $i$ . Clips with excessive drift are later removed in Sec. A.3, ensuring consistent gravity-aware orientations.

**Quality Filtering.** The raw 360° videos contain various artifacts, which we address using three complementary filtering strategies:

- **Low-quality filtering:** We use Q-Align [63] to assess image quality, aesthetics, and video quality, and average these scores as a quality indicator for candidate selection in Sec. A.3.
- **Watermark filtering:** A watermark detector [33] is applied to each frame, and the averaged score is used as a watermark indicator for candidate selection.
- **vLLM-based filtering:** A custom vLLM-based filter [4] (see Fig. A.1) identifies and removes clips that are non-equirectangular, low-quality, or contain overlays, missing edges, or other artifacts. The vLLM model also assigns each clip to most relevant POI (Point of Interest) categories [64] for later semantic balancing.

**Trajectory Scale Normalization.** Monocular SLAM recovers camera trajectories only up to an arbitrary scale, causing the same unit translation to appear faster in shallow scenes and slower in deeper ones. To ensure consistent motion across clips, we estimate a per-clip geometric scale and normalize all trajectories accordingly. Specifically, we compute dense optical flow using PanoFlow [51], recover per-pixel depth via generalized epipolar geometry, and obtain a near-plane statistic by taking the 25th percentile of valid depths per frame and then the median across the clip. We rescale each trajectory by this value so that the median near-plane depth becomes 1, yielding consistent apparent motion across all training clips.

### A.3. Emulating Realistic Camera Rotations

The curated panoramic clips serve as exploration spaces with provided camera translations, but determining realistic camera rotations remains challenging. Synthetic rotations (e.g., constant-speed sweeps) appear unnatural and do not reflect real panning, tilting, or rolling patterns that correlate with translation.

Our key insight is that videos with similar translation motions tend to exhibit similar rotation dynamics. Thus, we employ a matching-based approach with two steps: (1) extraction of realistic rotation trajectories from perspective videos, and (2) matching and transferring these rotations to panoramic clips based on translation similarity and clip quality.

**Candidate Rotation Extraction.** We use CameraBench [37], a dataset of 1k motion-diverse cinematic videos with text-described camera motions. We remove clips with “zoom” in descriptions to maintain fixed intrinsics, and extract camera poses using ViPE [24], producing around 300 unique trajectories. In some challenging scenarios, ViPE may produce unstable or jittery poses. To remove these results, we compute each clip’s maximum rotational velocity and discard the top 20%. We then align all trajectories to a gravity-aware world frame via GeoCalib [57] by aligning the first-frame up vector to the world up axis. This step ensures that all candidate rotations can be directly applied to the gravity-aligned panoramic clips to produce natural absolute orientations.

**Trajectory Matching and Rotation Transfer.** For each panoramic clip, we find CameraBench trajectories with similar translations and transfer their rotations to the virtual camera. We first align each candidate trajectory to the panoramic clip via Umeyama fitting [54], constrained to vertical-axis rotation. Then we retain the top 30 candidates with lowest RMSE error for further selection.

The original 360° videos exhibit serious long tails in POI categories (see Fig. A.2a), possibly due to the widespread use of 360° cameras for specific scenarios such as outdoor sports in the mountains and street views. Therefore we apply a series of diversity constraints during candidate selection to ensure semantic balance and motion diversity. Specifically, each candidate is scored using a composite metric averaging quality score, watermark score, and the rotation score  $\alpha_{\max}$  from Sec. A.2. Then they are traversed greedily from best to worst while enforcing the following diversity constraints:

- **Panoramic clip diversity:** At most 5 CameraBench matches per panorama.
- **CameraBench trajectory diversity:** Each trajectory can be used at most 100 times.
- **POI semantic balance:** Matches are skipped if all POI categories of the panoramic clip already exceed 1000 matches.
- **Motion diversity:** We compute the average optical-flow magnitude as a motion score and maintain bins of fixed width of 10, each capped at 2000 samples.

While these constraints do not guarantee perfect uniformity, they substantially reduce long-tail distributions in POI categories (see Fig. A.2b) and motion. After selection, each panoramic clip is paired with 5 per-frame camera rotations aligned to equirectangular coordinate system, yielding a total of 12k pairs of panoramic clips and rotation sequences.

### A.4. Video Synthesis with Diverse Cameras

From the previous stage, each panoramic clip is paired with five transferred rotation trajectories. To further enrich mo-

You are a video understanding assistant specialized in analyzing panoramic ERP-format videos. Given one frame of a panoramic video, your tasks are:

- Filtering**: Identify if the video should be filtered out. Output boolean flags for the following conditions (true if the issue exists, false otherwise):
  - `non_erp_format`: The video is **not** in ERP (Equirectangular Projection) panoramic format. For example, if the video looks like a flat perspective, fisheye, cube-map, or any projection other than ERP, set this to true.
  - `has_subtitle_or_watermark`: The video contains **text overlays, subtitles, logos, or watermarks**. Look carefully for visible text at the bottom, center, or corners of the video. If such elements are present and not part of the real scene, set this to true.
  - `edge_missing`: The top or bottom edges of the ERP panorama are **cut off, blacked out, cropped, or covered by logos/watermarks**, so the full 360 vertical coverage is missing or obstructed. If you cannot clearly see the poles (sky/ground) or if the edges are hidden by overlays, set this to true.
  - `has_overlay`: The frame contains **artificial overlays**, such as embedded UI elements, pop-up graphics, stickers, video-in-video inserts, menus, or other synthetic elements that are not part of the natural scene. If you see signs of AR/VR interface, streaming UI, or added images, set this to true.
  - `low_quality`: The video is of **poor visual quality**, such as being blurry, noisy, heavily pixelated, very low resolution, or distorted in a way that prevents recognizing the scene. If the content is hard to interpret due to quality issues, set this to true.
  - `unnatural_content`: The video contains **cartoons, animations, CGI, synthetic 3D renderings, or game engine graphics** rather than real-world panoramic footage. If the content is not realistic, set this to true.
- POI Categorization**: From the provided list of categories, select **one or more most relevant** labels that best describe the scene. Only use the given categories, do not invent new ones.

---

**poi\_category list (choose only from below):**

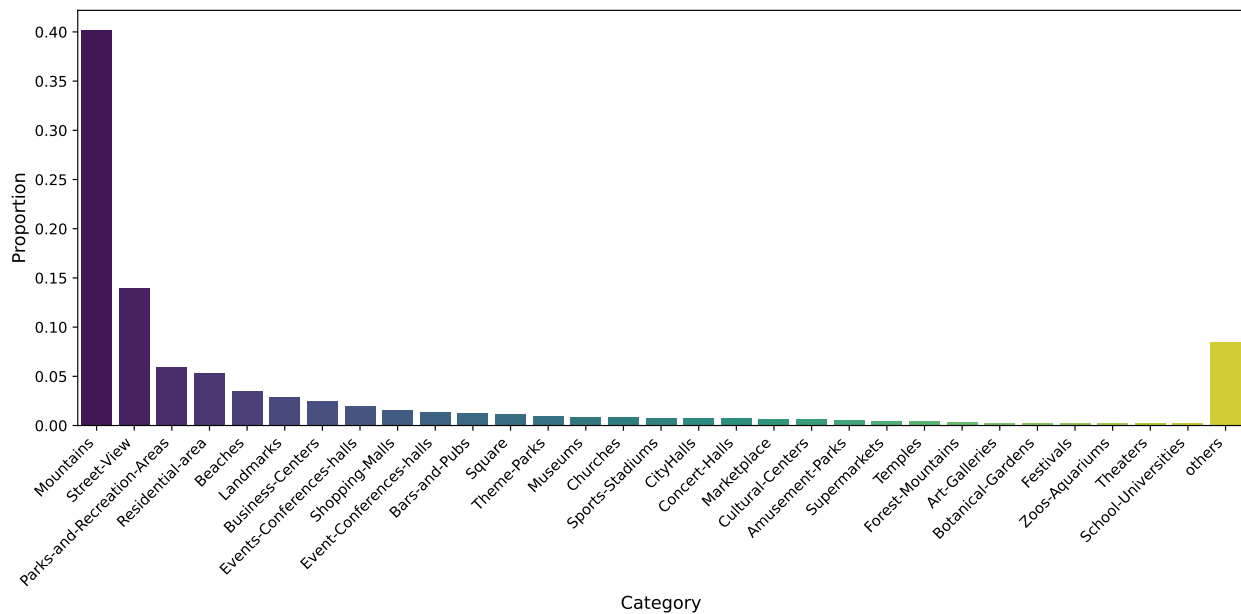
Restaurant, Coffee-Shop, Bars-and-Pubs, Residential-area, Hotels-Motels, Vacation-Rentals, Hospitals-Clinics, Pharmacies, Dentists, School-Universities, Library, Supermarkets, Shopping-Malls, Clothing-Stores, Shoe-Stores, Bookstores, Flowerstore, Furniture-Stores, Electorical-Store, Pet-Store, Toy-Shop, Airports, Train-Stations, Bus-Stops, Gas-Station, Car-Rental-Agencies, Theaters, Concert-Halls, Sports-Stadiums, Parks-and-Recreation-Areas, Museums, Art-Galleries, Zoos-Aquariums, Botanical-Gardens, Landmarks, Cultural-Centers, Post-Offices, Police-Stations, Courthouses, CityHalls, Banks-ATMs, Events-Conferences-halls, Beaches, Hiking-Trails, Campgrounds, Lakes, Mountains, Forest-Mountains, Farms, Street-View, Square, Business-Centers, Tech-Companies, Co-working-Spaces, Gyms-and-Fitness-Centers, Sports-Clubs, Swimming-Pools, Tennis-Courts, Auto-Repair-Shops, Car-Washes, Parking-Lots, Churches, Mosques, Temples, Graveyards.

---

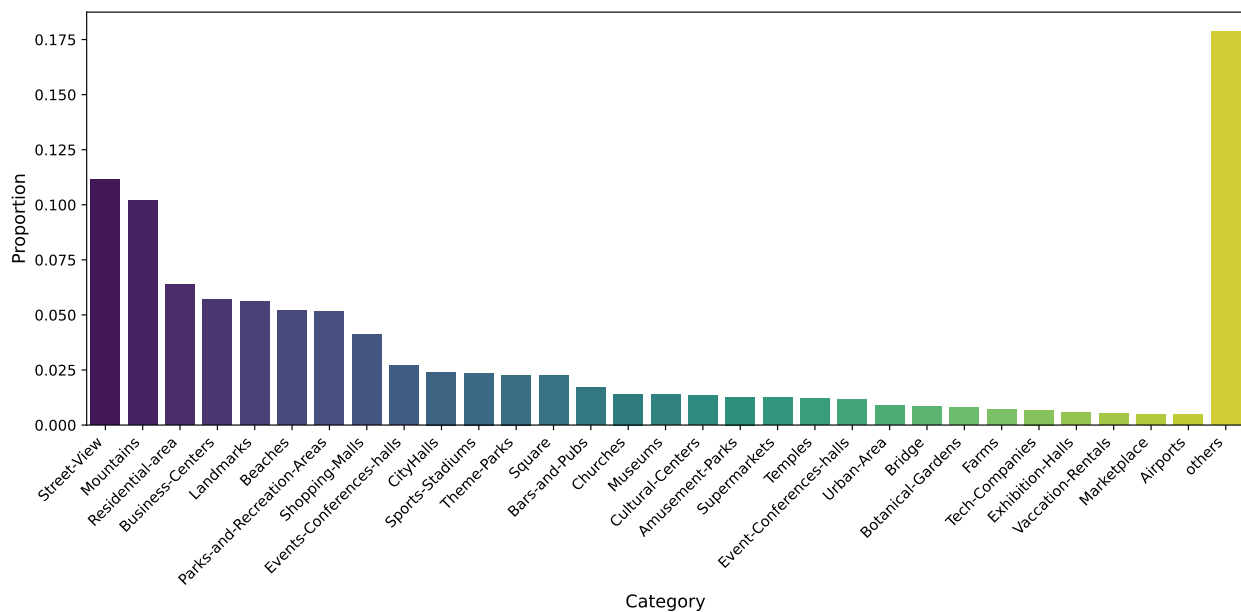
**Output strictly in JSON format** as follows:

```
```json
{
  "filter": {
    "non_erp_format": false,
    "has_subtitle_or_watermark": false,
    "edge_missing": false,
    "has_overlay": false,
    "low_quality": false,
    "unnatural_content": false
  },
  "poi_category": ["Mountains"]
}
```
```

Figure A.1. Prompt used for panoramic video filtering and scene categorization.



(a) POI category distribution before diversity-constraint trajectory matching.



(b) POI category distribution after diversity-constraint trajectory matching.

**Figure A.2. Effect of diversity constraints on POI category distribution.** The original 360° panoramic clips exhibit a highly long-tailed distribution across POI categories, dominated by scenes such as mountains and street views (a). After applying our diversity constraints during trajectory matching by limiting clip/trajectory reuse and enforcing semantic balance, the resulting dataset becomes more balanced (b), reducing over-represented categories and improving semantic coverage for downstream video synthesis.

tion variation, we apply additional perturbations and panning motions to these trajectories. We then use the Unified Camera Model (UCM) [41] to synthesize camera-diverse videos by projecting the panoramic frames into virtual cameras with varied intrinsics and distortions.

**Camera Rotation Augmentation.** For each transferred trajectory, we generate three augmented variants:

- **Consistent yaw perturbation:** A random yaw offset sampled from  $[-180^\circ, 180^\circ]$  is added to all frames.
- **Consistent yaw/pitch perturbation:** Random yaw and pitch offsets sampled from  $[-180^\circ, 180^\circ]$  and  $[-80^\circ, 80^\circ]$ , respectively, are applied uniformly across the clip.
- **Smooth panning motion:** A smooth yaw-pitch-roll panning curve with random start and end offsets is added over the clip duration, using sampling ranges of  $[-90^\circ, 90^\circ]$  for yaw,  $[-40^\circ, 40^\circ]$  for pitch, and  $[-30^\circ, 30^\circ]$  for roll.

These augmentations produce three additional rotation variants per trajectory, resulting in a total of 48k unique camera motions. Each augmented rotation sequence defines a virtual camera rotation  $\mathbf{R}^{\text{aug}} \in \text{SO}(3)$  for each panoramic frame, used later to project into the desired view.

**Unified Camera Model (UCM).** To synthesize videos with diverse intrinsics and distortions, we adopt the Unified Camera Model (UCM) [41], which represents a wide range of central cameras using a single distortion parameter  $\xi$ .

**Projection.** Given a 3D point  $\mathbf{p} = [p_x, p_y, p_z]^\top$  in the camera coordinate system, UCM first computes:

$$r = \sqrt{p_x^2 + p_y^2 + p_z^2}, \quad \beta = p_z + \xi r, \quad (\text{A.3})$$

and then projects the point to pixel coordinates:

$$u = f_x \frac{p_x}{\beta} + c_x, \quad v = f_y \frac{p_y}{\beta} + c_y, \quad (\text{A.4})$$

where  $f_x, f_y, c_x, c_y$  denote the camera intrinsics. The distortion parameter  $\xi$  controls the nonlinearity of the projection: when  $\xi = 0$ , UCM reduces to the standard pinhole model, while larger values of  $\xi$  introduce stronger bending of rays, allowing UCM to approximate wide-angle and fish-eye lenses within a unified formulation.

**Ray mapping.** For a pixel coordinate  $(u, v)$ , we first obtain normalized image coordinates:

$$x = \frac{u - c_x}{f_x}, \quad y = \frac{v - c_y}{f_y}, \quad (\text{A.5})$$

and then lift  $(x, y)$  back to a 3D ray direction under UCM:

$$\mathbf{d}^{\text{cam}} = \frac{1}{\sqrt{x^2 + y^2 + (1 - \xi\rho)^2}} \begin{bmatrix} x \\ y \\ 1 - \xi\rho \end{bmatrix}, \quad (\text{A.6})$$

$$\rho = \sqrt{x^2 + y^2}.$$

Since UCM is central, each ray originates from the same center  $\mathbf{o}^{\text{cam}} = \mathbf{0}$ .

**FoV-based intrinsic re-parameterization.** Instead of specifying intrinsics through  $f_x$  and  $f_y$ , which is less intuitive, we re-parameterize the camera using its horizontal field of view xFoV. For an image width  $W$ , the corresponding focal length is:

$$f_x = f_y = \frac{W}{2} \frac{\cos \gamma + \xi}{\sin \gamma}, \quad \gamma = \frac{1}{2} \text{xFoV}. \quad (\text{A.7})$$

Throughout our formulation, we assume the principal point is centered in the image,  $c_x = \frac{1}{2}W$ ,  $c_y = \frac{1}{2}H$ , where  $H$  and  $W$  denote the image height and width.

**Final ray mapping and projection.** Using the focal length derived from Eq. (A.7), the UCM unprojection in Eq. (A.6) defines, for each pixel  $(u, v)$ , a central-camera ray

$$\mathbf{d}_{u,v}^{\text{cam}} = \Phi^{\text{UCM}}(u, v; \text{xFoV}, \xi, H, W), \quad (\text{A.8})$$

where  $\Phi^{\text{UCM}}$  denotes the FoV-parameterized UCM ray-mapping function.

Analogously, the UCM projection of a 3D point  $\mathbf{p} \in \mathbb{R}^3$  to the image plane is given by another mapping

$$(u, v) = \Pi^{\text{UCM}}(\mathbf{p}; \text{xFoV}, \xi, H, W), \quad (\text{A.9})$$

which applies Eq. (A.4) using the same FoV-parameterized intrinsics. These two operators form the basis of UCM and are used throughout the pipeline to generate camera-diverse views.

**Camera Intrinsics and Distortion Sampling.** To cover a broad range of camera geometries, we sample the horizontal field of view xFoV and UCM distortion parameter  $\xi$  from several lens-dependent uniform ranges. We organize cameras into four categories, from pinhole to extreme fish-eye, and draw xFoV and  $\xi$  uniformly within their respective intervals:

- **Pinhole:**  $\text{xFoV} \in [90^\circ, 110^\circ]$ ,  $\xi \in [0.0, 0.0]$ .
- **Wide-angle:**  $\text{xFoV} \in [110^\circ, 140^\circ]$ ,  $\xi \in [0.5, 0.95]$ .
- **Fisheye:**  $\text{xFoV} \in [140^\circ, 180^\circ]$ ,  $\xi \in [1.05, 2.0]$ .
- **Extreme fisheye:**  $\text{xFoV} \in [160^\circ, 200^\circ]$ ,  $\xi \in [1.5, 2.3]$ .

This sampling scheme provides broad coverage of real-world intrinsics and distortion levels, enabling UCPE to learn consistent ray representations across diverse camera types.

**Panoramic-to-UCM Projection.** Given an equirectangular panorama  $I^{\text{ERP}} \in \mathbb{R}^{H' \times W' \times 3}$ , our goal is to project it to a UCM virtual camera view by mapping each UCM ray to its corresponding location on the panorama.

For each pixel  $(u, v) \in \{1, \dots, W\} \times \{1, \dots, H\}$ , we first obtain the FoV-parameterized UCM ray  $\mathbf{d}_{u,v}^{\text{cam}}$  using Eq. (A.8). Applying the virtual camera rotation  $\mathbf{R}^{\text{aug}} \in \text{SO}(3)$  yields the ray expressed in the equirectangular camera frame:

$$\mathbf{d}_{u,v}^{\text{ERP}} = \mathbf{R}^{\text{aug}} \mathbf{d}_{u,v}^{\text{cam}}. \quad (\text{A.10})$$

Denoting ray as  $\mathbf{d}_{u,v}^{\text{ERP}} = [d'_x, d'_y, d'_z]^\top$ , its corresponding location on the panorama is obtained using spherical projection:

$$(u', v') = \left( \frac{\text{atan2}(d'_x, d'_z) + \pi}{2\pi}, \frac{\arcsin(d'_y) + \frac{\pi}{2}}{\pi} \right). \quad (\text{A.11})$$

The final UCM-rendered frame is obtained by sampling the panorama at these coordinates:

$$I^{\text{UCM}}[u, v] = I^{\text{ERP}}(u', v'), \quad (\text{A.12})$$

resulting in a UCM image consistent with the sampled intrinsics  $(x\text{FoV}, \xi)$  and the virtual camera pose. This process is repeated for all frames in the panoramic clip to produce the final synthesized video.

**Virtual Camera Pose Composition.** For each rendered frame, the equirectangular SLAM system provides a camera-to-world pose  $\mathbf{T}^{\text{ERP}} \in \text{SE}(3)$ , which we compose with the augmented rotation  $\mathbf{R}^{\text{aug}} \in \text{SO}(3)$  to obtain the final virtual-camera pose. Let  $\mathbf{R}^{\text{ERP}}$  and  $\mathbf{t}^{\text{ERP}}$  denote the rotation and translation of  $\mathbf{T}^{\text{ERP}}$ . The virtual UCM pose is then

$$\mathbf{T}^{\text{UCM}} = \begin{bmatrix} \mathbf{R}^{\text{ERP}} \mathbf{R}^{\text{aug}} & \mathbf{t}^{\text{ERP}} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad (\text{A.13})$$

which replaces only the orientation while preserving the original trajectory translation. This pose is paired with the sampled intrinsics  $(x\text{FoV}, \xi)$  for camera-aware training.

**UCM Video Captioning.** To provide descriptive captions for each synthesized video for text-to-video training, we adapt vLLM model [4] with a simple prompt:

You are a helpful video captioning assistant. Please describe this video in detail.

In total, we generate approximately 48k 81-frames video clips at a resolution of  $480 \times 832$  and 16 fps. The entire generation process takes approximately 3 days on a single A800 GPU node. For evaluation, we match the official test split of CameraBench with the rest of  $360^\circ$  videos and enforce stricter diversity constraints, resulting in 272 test clips. We note that while our dataset is synthesized with UCM, our UCPE representation is compatible with other camera models, as it encodes rays in a model-agnostic manner.

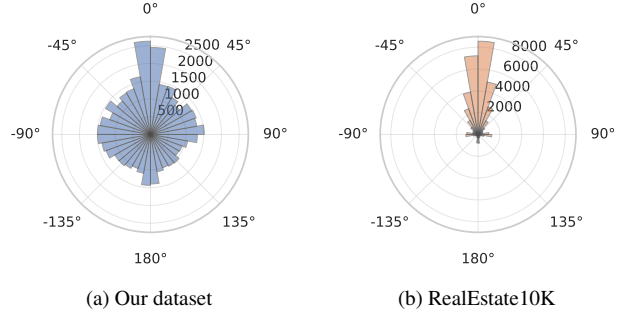


Figure A.3. **Camera Motion Direction Distribution.** We visualize the horizontal translation directions using rose plots for (a) our dataset and (b) RealEstate10K. Compared with the strong forward-motion bias in RealEstate10K, our dataset provides substantially richer and more uniformly distributed camera translation directions.

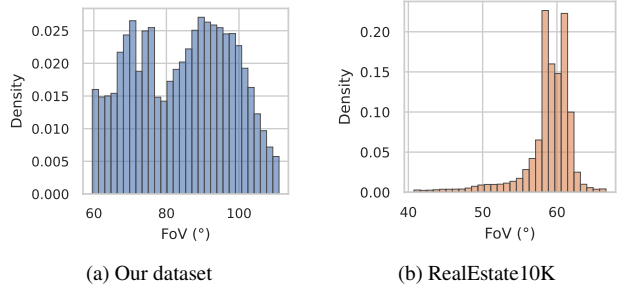


Figure A.4. **Camera Field of View Distribution.** We compare the vertical FoV distributions of (a) our dataset and (b) RealEstate10K. Our dataset exhibits a wide FoV range ( $60^\circ$ - $110^\circ$ ), reflecting the diverse intrinsics sampled during synthesis. In contrast, RealEstate10K shows a narrow distribution concentrated around  $60^\circ$ , indicating limited variability in camera intrinsics.

## A.5. Camera Motion and Intrinsics Statistics

**Camera Motion Direction Distribution.** To assess the diversity of camera motion in our dataset, we visualize the distribution of translation directions between the first and last frames, as shown in Fig. A.3. We compare our synthetic dataset against RealEstate10K [77], a real-world collection containing approximately 66k clips in the training split. For fair comparison, we discard clips with fewer than 81 frames and use the first 81 frames of the remaining videos, yielding 47k clips. As shown in Fig. A.3a and Fig. A.3b, our dataset covers a more balanced range of translation directions, largely due to the consistent yaw perturbation introduced during augmentation, whereas RealEstate10K is dominated by forward translations characteristic of real-estate tours.

**Field of View Distribution.** We further compare the vertical field-of-view (FoV) distributions of our dataset

and RealEstate10K. As shown in Fig. A.4, our dataset spans a broad FoV range from  $60^\circ$  to  $110^\circ$ , owing to the diverse intrinsics sampled during synthesis. In contrast, RealEstate10K is largely concentrated near  $60^\circ$ , reflecting its limited intrinsic variability. Additionally, our dataset incorporates a wide spectrum of distortion levels under the Unified Camera Model (UCM), covering multiple lens types, whereas RealEstate10K primarily contains distortion-free pinhole cameras.

## B. Details of Absolute Orientation Encoding

As introduced in Sec. 3.2 of the main paper, the Lat-Up representation consists of a latitude map and an Up map. The latitude component is computed directly from world-space ray directions using Eq. 8 of the main paper. This section details the derivation of the Up map.

For each token  $t$ , let  $(u_t, v_t)$  denote its pixel center in the image plane. Using the FoV-parameterized UCM intrinsics collected in  $\phi$ , the camera-frame ray direction is obtained via the UCM ray-mapping operator defined in Eq. (A.8):

$$\mathbf{d}_t^{\text{cam}} = \Phi_\phi^{\text{UCM}}(u_t, v_t), \quad \|\mathbf{d}_t^{\text{cam}}\| = 1. \quad (\text{B.1})$$

Applying the camera-to-world rotation  $\mathbf{R}$  yields the world-frame ray direction (Eq. 1 in the main paper):

$$\mathbf{d}_t = \mathbf{R} \mathbf{d}_t^{\text{cam}}, \quad \|\mathbf{d}_t\| = 1. \quad (\text{B.2})$$

**Ray perturbation direction towards world up.** The Up map is constructed by examining how each viewing ray  $\mathbf{d}_t$  responds to an infinitesimal perturbation toward the world up axis  $\mathbf{u}^{\text{wld}} = [0, -1, 0]^\top$ . Intuitively, this perturbation reflects how the ray would move on the unit viewing sphere if it were slightly rotated in the direction of world up, and its image-plane displacement reveals the 2D Up direction associated with token  $t$ .

On the sphere, the instantaneous motion of the ray under such a perturbation must lie in the tangent direction orthogonal to both  $\mathbf{d}_t$  and  $\mathbf{u}^{\text{wld}}$ . This tangent direction is given by the cross product:

$$\mathbf{k}_t = \mathbf{d}_t \times \mathbf{u}^{\text{wld}}, \quad \hat{\mathbf{k}}_t = \frac{\mathbf{k}_t}{\|\mathbf{k}_t\|}. \quad (\text{B.3})$$

The unit vector  $\hat{\mathbf{k}}_t$  thus specifies the correct axis for applying an infinitesimal rotation to  $\mathbf{d}_t$  toward world up, forming the basis for the Up map defined in the following steps.

**Small-angle perturbation via Rodrigues’ formula.** We rotate  $\mathbf{d}_t$  by a small fixed angle  $\delta$  (e.g.,  $\delta = 0.1$  rad) around  $\hat{\mathbf{k}}_t$ . Using Rodrigues’ formula, the perturbed world-space

ray direction is

$$\mathbf{d}_t^{\text{rot}} = \text{Rot}(\hat{\mathbf{k}}_t, \delta) \mathbf{d}_t \quad (\text{B.4})$$

$$= \mathbf{d}_t \cos \delta + (\hat{\mathbf{k}}_t \times \mathbf{d}_t) \sin \delta + \hat{\mathbf{k}}_t (\hat{\mathbf{k}}_t^\top \mathbf{d}_t) (1 - \cos \delta). \quad (\text{B.5})$$

We then transform this perturbed ray back to the camera frame:

$$\mathbf{d}_t^{\text{cam,rot}} = \mathbf{R}^\top \mathbf{d}_t^{\text{rot}}, \quad (\text{B.6})$$

and project it with the UCM projection operator  $\Pi_\phi^{\text{UCM}}$  introduced in Eq. (A.9):

$$(u_t^{\text{rot}}, v_t^{\text{rot}}) = \Pi_\phi^{\text{UCM}}(\mathbf{d}_t^{\text{cam,rot}}). \quad (\text{B.7})$$

Importantly, the original  $(u_t, v_t)$  is the pixel location of token  $t$ , while  $(u_t^{\text{rot}}, v_t^{\text{rot}})$  is the projected position of the ray after a small rotation toward the world up direction.

**Up map definition.** The induced image-plane displacement is

$$\Delta u_t = u_t^{\text{rot}} - u_t, \quad \Delta v_t = v_t^{\text{rot}} - v_t, \quad (\text{B.8})$$

and the Up map at token  $t$  is defined as

$$\text{Up}_t = \frac{[\Delta u_t, \Delta v_t]}{\sqrt{\Delta u_t^2 + \Delta v_t^2}}. \quad (\text{B.9})$$

Beyond its geometric definition, the resulting Up map provides a strong appearance-dependent cue: its spatial pattern varies coherently with the camera’s absolute pitch and roll, and also responds to scene semantics such as the vertical structure of buildings and trees. Moreover, because the perturbation is evaluated through the projection function  $\Pi$ , the Up map naturally reflects the characteristic warping of different lenses. As a result, it offers a unified visual signal that jointly encodes global orientation, semantic regularities, and lens-induced distortions.

## C. Evaluation Metrics

In Sec. 4.1 of the main paper, we evaluate both video generation quality and camera controllability. Here, we detail the computation of camera-related metrics used for benchmarking relative pose control, absolute orientation, and lens controllability.

**Relative Camera Pose Control.** For each generated video, we begin by rectifying every frame to a pinhole projection. This step is necessary because existing pose estimation methods, including ViPE [24], are not robust under strong lens distortions and wide-FoV warping, which

frequently appear in our synthesized UCM views. Using the ground-truth UCM parameters ( $x\text{FoV}, \xi$ ), each distorted frame is mapped to a pinhole image whose effective horizontal FoV is capped at  $100^\circ$ . Rectification is implemented using the UCM ray-pixel operators  $\Phi_\phi^{\text{UCM}}$  and  $\Pi_\phi^{\text{UCM}}$ , which provide a forward-inverse mapping between distorted pixels and their underlying 3D rays. The rectified sequence is then fed to ViPE to estimate camera-to-world trajectories  $\{\hat{\mathbf{T}}_i^{\text{wc}}\}$ .

Following [19, 62], all pose metrics operate on *relative* trajectories. For a camera-to-world pose  $\mathbf{T}_i^{\text{wc}} = \begin{bmatrix} \mathbf{R}_i^{\text{wc}} & \mathbf{t}_i^{\text{wc}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in SE(3)$ , with rotation  $\mathbf{R}_i$  and translation  $\mathbf{t}_i$ , the corresponding relative ground truth pose and estimated pose is defined as

$$\mathbf{T}_i = (\mathbf{T}_0^{\text{wc}})^{-1} \mathbf{T}_i^{\text{wc}}, \quad \hat{\mathbf{T}}_i = (\hat{\mathbf{T}}_0^{\text{wc}})^{-1} \hat{\mathbf{T}}_i^{\text{wc}}. \quad (\text{C.1})$$

- **Rotation Error (RotErr):** The total rotation error is computed as the sum of per-frame angular deviations:

$$\text{RotErr} = \sum_i \arccos\left(\frac{\text{Tr}(\mathbf{R}_i^\top \hat{\mathbf{R}}_i) - 1}{2}\right). \quad (\text{C.2})$$

- **Translation Error (TransErr):** For relative translations  $\mathbf{t}_i$  and  $\hat{\mathbf{t}}_i$ ,

$$\text{TransErr} = \sum_i \|\mathbf{t}_i - \hat{\mathbf{t}}_i\|_2. \quad (\text{C.3})$$

- **Camera Motion Consistency (CamMC):** Flatten first three rows of each relative pose into a 12-dim vector and compute

$$\text{CamMC} = \sum_i \|\text{Vec}(\mathbf{T}_i) - \text{Vec}(\hat{\mathbf{T}}_i)\|_2. \quad (\text{C.4})$$

These metrics evaluate how closely the generated trajectory follows the target camera motion, independent of absolute scale or global alignment.

Since different baselines generate videos with varying lengths and frame rates (*e.g.*, CameraCtrl produces 16 frames at 4 fps, AC3D produces 48 frames at 8 fps, while our method outputs 81 frames at 16 fps), a direct trajectory comparison would be biased by temporal sampling. To ensure a fair evaluation of relative camera pose control, we uniformly sample the same 16 timestamps across all methods to compute the pose metrics.

**Absolute Camera Orientation and Lens Control.** We employ GeoCalib [57] to estimate per-frame absolute orientation and lens parameters from generated frames. For each frame  $i$ , GeoCalib outputs the predicted pitch  $\hat{\theta}_i$  and roll  $\hat{\varphi}_i$ , the vertical field of view  $\hat{\text{FoV}}$ , and the radial distortion coefficients  $\hat{k}_1, \hat{k}_2$  under the classical radial model [8].

Ground-truth parameters ( $\theta_i^*, \varphi_i^*, \text{FoV}^*, k_1^*, k_2^*$ ) are obtained from the GeoCalib estimation on the original UCM frames. We then compute the following metrics:

- **Pitch / Roll Error:** We compute the absolute error

$$|\hat{\theta}_i - \theta_i^*|, \quad |\hat{\varphi}_i - \varphi_i^*|. \quad (\text{C.5})$$

- **FoV and Distortion Errors:** Under the radial Brown model [8], we compute

$$|\hat{\text{FoV}} - \text{FoV}^*|, \quad |\hat{k}_1 - k_1^*|, \quad |\hat{k}_2 - k_2^*|. \quad (\text{C.6})$$

Together, these metrics measure controllability over absolute camera orientation, lens intrinsics, and distortion, providing a detailed evaluation of camera-aware generation quality.

## D. Implementation Details

**Our Method.** We adopt Wan2.1-T2V-1.3B [59] as our base model. It consists of a text encoder, a Diffusion Transformer, and a 3D VAE, totaling 7.3 billion parameters. We fine-tune the model on our synthesized dataset while freezing all original weights, and train only the proposed attention adapters using AdamW [38] with a learning rate of  $1e-4$  for 10k steps. Training uses a batch size of 8 on 8 NVIDIA A800 GPUs and completes within roughly one day. At inference time, we generate 81 frames at a resolution of  $480 \times 832$  and 16 fps. Since no existing text-to-video model supports full camera control (*i.e.*, relative pose, absolute orientation, and lens parameters), we adapt two recent methods for comparison, as detailed below.

**ReCamMaster.** We adapt the official ReCamMaster implementation [3] for text-to-video generation with complete camera control. As illustrated in Fig. D.1a, ReCamMaster injects raw camera parameters into every Transformer block. In our adaptation, we concatenate the normalized field of view  $x\text{FoV}' = x\text{FoV}/180^\circ$  and the distortion parameter  $\xi$  with the flattened 12-dim pose vector  $\text{Vec}(\mathbf{T}_i^{\text{wc}})$  for each frame  $i$ . Because Wan2.1-T2V-1.3B applies a  $4\times$  temporal compression in the VAE, these per-frame parameters are downsampled to match the token sequence length  $n$ . Following the original design, the camera parameters are then broadcast across all spatial tokens ( $h \times w$ ), yielding a tensor of shape  $(nhw, 14)$ , projected with zero-initialized linear layers to the token dimension, and added as a bias before each self-attention layer. A linear layer with identity initialization performs feature projection after self-attention. During training, we fine-tune the added linear layers and the original self-attention layers following the ReCamMaster training scheme.

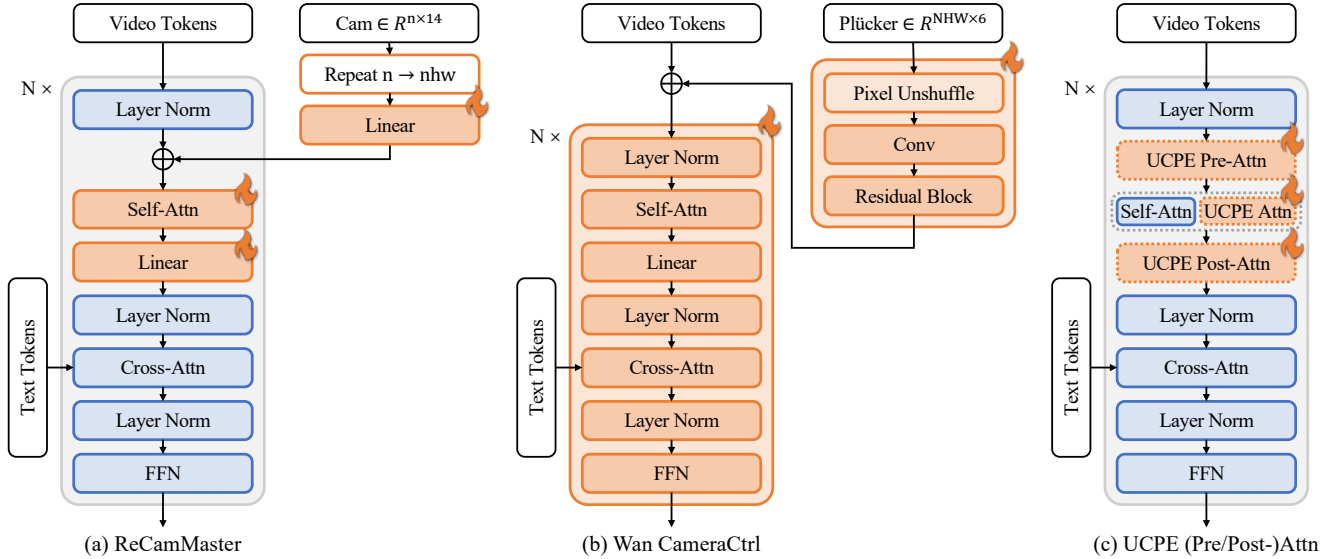


Figure D.1. **Implementation of Baseline And Ablation Models.** (a) ReCamMaster injects per-frame camera parameters into each Transformer block after spatial repetition. (b) Wan CameraCtrl injects Plücker-encoded rays into video tokens using a convolutional adapter. (c) Our UCPE ablations insert a spatial attention adapter before, after, or in parallel with the original self-attention module.

**Wan CameraCtrl.** We further adapt Wan2.1-Fun-V1.1-1.3B-Control-Camera, a third-party implementation of CameraCtrl [19] built on Wan [59], for text-to-video generation. As shown in Fig. D.1b, Wan CameraCtrl converts Plücker-encoded rays of shape  $(NHW, 6)$ , where  $N$  is the number of frames and  $H, W$  denote spatial resolution of input video, into token-shaped features using a convolutional adapter with pixel-unshuffle downsampling, and injects them via additive biasing. We retain this structure but generalize the Plücker encoding from pinhole cameras to UCM cameras using the ray formulation in Eq. (A.8). For stable training, the convolution and final residual layers are initialized to zero. We fine-tune all parameters of the convolutional adapter and the diffusion transformer using a learning rate of  $1e-5$ , following the official training recipe.

**UCPE Spatial Adapters Ablation.** As described in Sec. 3.3 of the main paper, we insert a spatial attention adapter as a parallel branch to each Diffusion Transformer self-attention layer (see Fig. 3 of the main paper). In Sec. 4.3, we additionally evaluate two variants that place the adapter immediately before or after the original self-attention. The three variants are summarized in Fig. D.1c and share the same 1/8 compression ratio. All models are trained under the same setting described above.

**Inference Latency.** We measure the inference latency of UCPE and other baselines on an NVIDIA A800 GPU, as summarized in Tab. D.1. UCPE introduces minimal computational overhead compared to the base Wan2.1 model

Table D.1. **Inference latency comparison.** Latencies are measured on a single NVIDIA A800 GPU, with corresponding frame counts and resolutions provided for reference.

| Method         | Time (s) | Frames | Resolution       |
|----------------|----------|--------|------------------|
| ReCamMaster    | 179      | 81     | $480 \times 832$ |
| Wan CameraCtrl | 177      | 81     | $480 \times 832$ |
| CameraCtrl     | 17       | 16     | $256 \times 384$ |
| AC3D           | 632      | 48     | $480 \times 720$ |
| UCPE           | 184      | 81     | $480 \times 832$ |

and other camera-conditioned adapters, while generating high-resolution videos with a significantly larger number of frames than earlier methods such as CameraCtrl and AC3D. Specifically, UCPE generates 81 frames at  $480 \times 832$  resolution in 184 seconds, which is comparable to ReCamMaster (179s) and Wan CameraCtrl (177s) under the same settings.

## E. Robustness, Generalization, Limitations, and Future Works

**Robustness to Text-Camera Conflict.** We further evaluate UCPE’s robustness when textual prompts conflict with the specified camera geometry. As shown in Fig. E.1, we test a prompt specifying a “flat, telephoto portrait” of a cat against a contradictory fisheye camera ( $x\text{FoV} = 180^\circ, \xi = 2.0$ ). Despite the text suggesting a narrow field of view and minimal distortion, UCPE successfully enforces the requested fisheye geometry, demonstrating that the camera positional encoding provides strong geometric guidance that can override conflicting appearance cues in the text.



Figure E.1. **Robustness to Text-Camera Conflict.** Even when the text prompt specifies a “telephoto” view, UCPE correctly synthesizes a fisheye video according to the provided camera parameters, resolving the conflict between text and geometry.



Figure E.2. **Generalization to Unseen Camera Models.** Without fine-tuning, UCPE generalizes to unobserved models such as the Brown-Conrady model by simply substituting the ray-mapping function at inference time.



Figure E.3. **Failure Case on Equirectangular Projection.** Testing with ERP ray-mapping results in artifacts as the model was not trained on such extreme panoramic projections.

**Generalization to Unseen Camera Models.** UCPE is model-agnostic as it learns universal ray-space interactions. For unobserved lens types, one can simply replace the ray-mapping function at inference to achieve control without fine-tuning, as the attention mechanism reasons about ray relationships rather than specific camera parameters (*e.g.*,  $\xi$ ). We evaluate this capability on the complex Brown-Conrady camera model used in OpenCV for physical lens calibration ( $x\text{FoV} = 100^\circ$ ,  $k_1 = -0.2$ ,  $k_2 = -0.05$ ). As shown in Fig. E.2, UCPE successfully generalizes to this model, adding realistic barrel distortion to the synthesized frames without requiring additional training.

**Limitations.** UCPE relies on camera poses during training and currently models only pose, intrinsics, and distortion, without capturing richer attributes such as zoom, focus, or depth-of-field. Furthermore, we test UCPE with the ray-mapping function of Equirectangular projection (ERP), representing an extreme case not seen during training. As shown in Fig. E.3, while UCPE fails to synthesize full  $360^\circ$  panoramas correctly under this extreme setting, we expect its performance to improve with targeted fine-tuning on panoramic datasets.

**Future Works.** Extending UCPE to support these additional controls and further reducing reliance on accurate pose supervision (*e.g.*, through self-supervised geometric losses) remain promising directions. Furthermore, while



Figure E.4. **Application to I2V Tasks.** We fine-tune Wan2.1-I2V-14B to demonstrate the feasibility of UCPE on more constrained Image-to-Video tasks.

T2V generation is our primary focus, UCPE is also applicable to Image-to-Video (I2V) tasks. In contrast to T2V, I2V tasks benefit from the strong rigid pixel constraints of the input image, which naturally anchors the lens and initial orientation. Consequently, UCPE supports the I2V task without the need for Absolute Orientation Encoding (AOE): once the lens is defined (*e.g.*, via GeoCalib), Relative Ray Encoding (RRE) alone can drive subsequent frames. To demonstrate this feasibility, we fine-tune Wan2.1-I2V-14B. As shown in Fig. E.4, our representations readily extend to control the camera pose in more constrained I2V settings. Video-to-Video (V2V) tasks (*e.g.*, ReCamMaster) could similarly benefit from UCPE’s enhanced controllability. Since such tasks inherently require synthetic multi-trajectory datasets, we leave this exploration to future work.

## F. Supplementary Video

The supplementary video provides visualizations and demonstrations of the following aspects:

- Overview of UCPE (Unified Camera Positional Encoding) and its two components: Relative Ray Encoding and Absolute Orientation Encoding.
- Demonstration of controllability over intrinsics and distortions.
- Demonstration of controllability over extrinsics and initial camera orientation.
- Applications enabled by UCPE, including cinematic content creation, autonomous driving, and embodied AI.
- Examples from our large-scale synthetic dataset with diverse intrinsics, distortion profiles, and camera motions (Sec. 4.1 of the main paper, Sec. A of the supplementary).
- Comparison with baseline methods on our synthetic dataset for complete camera-controlled video generation (Fig. 4 of the main paper).
- Generalization results on the RealEstate10K dataset, showcasing improved camera controllability and video quality under pinhole camera settings (Fig. 5 of the main paper).